UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

SCREW - SEMANTIC CONTENT ANALYSIS FOR REPAIR AND EVALUATION OF WEB ACCESSIBILITY

Inês Teixeira de Matos

Mestrado em Informática

Dissertação orientada por:
Prof. Doutor Carlos Alberto Pacheco dos Anjos Duarte
e Prof. Doutor Luís Manuel Pinto da Rocha Afonso Carriço

2017

Acknowledgments

Firstly, I would like to thank my teachers, Prof. Carlos Duarte and Prof. Luís Carriço, for leading me to develop a great work and for helping me to improve myself in many ways, through their guidance and support.

I want to thank to the group in room 34 for the shared moments and for the good environment, even when everybody was feeling tired.

I can't forget my friends who followed me for all these years inside and outside of this institution, during this journey. Without you everything would be more difficult. You aren't just collegues that I met in school, you are the kind of people that I want to keep in my life. Specially, thank you Ana Salvado, João Vicente, Carlos Duarte, Tiago Moucho, Ana Espinheira and Pedro Luz for your presence and aid when I needed the most.

For my family and friends in long time, I'm grateful for all your support in keeping me up.

In last, thank you to all the people that answered and shared my surveys. Your cooperation was precious for my work. Thank you a lot.

“I have always believed, and I still believe, that whatever good or bad fortune may come our way we can always give it meaning and transform it into something of value."

— Herman Hesse in Siddartha

Resumo

A Internet tem continuamente vindo a ser integrada no nosso quotidiano, tanto num ambiente profissional, como num de entretenimento. Tornou-se um recurso importante para as nossas atividades diárias, desde o trabalho à recreação. Isto significa que cada vez mais as pessoas navegam na WWW. Contudo, existem muitos tipos de utilizadores e alguns deles sofrem de deficiências, restringindo a sua experiência de utilização. Isto leva a que haja uma procura por uma Web mais acessível para todos os tipos de utilizadores. No entanto, este processo seria mais difícil se não houvessem normas que recomendassem especificações para os sites seguirem e cumprirem, a fim de torná-los mais acessíveis. Felizmente, há uma organização designada pelas siglas WAI, Web Accessibility Initiative, que estabelece essas especificações como um conjunto de diretrizes (por exemplo, WCAG, Web Content Accessibility Guidelines), afim de ajudar no desenvolvimento das páginas web. Para ajudar os desenvolvedores, há também ferramentas como QualWeb, TotalValidator, entre outras, que permitem que os sites sejam avaliados de acordo com as diretrizes mencionadas acima, fornecendo resultados específicos. No entanto, a maioria destas ferramentas não obtém resultados com base na semântica de uma página e só conseguem fazer avaliações de sintaxe. Por exemplo, essas aplicações não avaliam se as descrições das imagens são realmente descritoras das mesmas. Nestes casos, a maioria das ferramentas pede ao desenvolvedor/utilizador para verificar manualmente. Além disso, nenhuma ferramenta conhecida consegue executar avaliações de acessibilidade Web e reparação automática. A reparação automática ajuda os utilizadores e programadores Web a navegar sem restrições, reparando no mesmo instante, e a transcrever de uma forma mais acessível o código, respetivamente. Assim, o principal tópico desta pesquisa é a análise de conteúdo Web semântico para melhorar a acessibilidade da Web e a sua reparação automática. Cada etapa de desenvolvimento, descrita nesta tese, será integrada no Qualweb, um avaliador de acessibilidade Web que pode realizar análise de conteúdo dinâmico.

Neste documento é apresentado, primeiramente, um estudo sobre as tecnologias e metodologias existentes para a avaliação semântica e reparação de código nas páginas Web e algumas noções necessárias para o entendimento do trabalho que foi realizado. É também descrito como funciona o Qualweb e a sua arquitetura, pelo que é a ferramenta principal a beneficiar deste estudo.

Relativamente ao trabalho, é apresentada uma ferramenta capaz de efetuar avaliações semânticas e geração de descrições sob conteúdo da Web, para fins de acessibilidade web, designada por Screw. Estes conteúdos irão corresponder a elementos de uma página

Web que, resumidamente, poderão ser conteúdos textuais, referências a imagens e elementos/atributos do DOM que descrevam estas informações. Desta forma irão haver dois tipos de entrada no sistema, o elemento a ser descrito e a sua descrição. Este elemento poderá ser textual ou uma imagem, no entanto para verificar a semalhança semântica entre dois tipos de contéudos diferentes (imagem e texto) é necessário converter a imagem para texto, através de interpretadores que oferecem um conjunto de conceitos, que de alguma forma descrevem a imagem. Após este processo, para cada conceito é retirada a relação semântica com a descrição e com um conjunto de domínios existentes no sistema e o mesmo acontece entre a descrição e os mesmos domínios. Estes domínios são uma componente importante do sistema, pois oferecem um conjunto de dados que contextualizam tanto os conceitos como a descrição. Isto é, se a descrição e um conceito estiverem semânticamente relacionados com um mesmo domínio, então existe uma probabilidade de estes dois estarem também semanticamente relacionados. Isto irá fortalecer a relação semântica entre o conteúdo a ser descrito e a descrição.

Após obter estes valores é aplicado um algoritmo que irá ditar se a descrição descreve ou não o contéudo. Para cada conceito e domínio existe, então, um valor semântico que os relaciona. Se a descrição tive algum valor relacional com esse mesmo domínio, então é aplicada uma fórmula de máximo entre, o valor da relação entre o conceito e o domínio e o valor da relação entre o conceito e a descrição. Após efetuar isto para todos os conceitos, é feita uma média de todos os valores acima de 0, isto é, que sejam positivos. Esta média irá corresponder à relação semântica entre a descrição e o conteúdo a ser descrito. Para além disto, é contado o número de vezes em que não existe relação entre os conceitos e a descrição, se este número estiver acima de um percentil, 0.6, então significa que não existe relação semântica entre a descrição e a imagem. A descrição é considerada boa ou má, se o valor obtido, pela média, está acima ou abaixo de um limite, 0.14. No fim, este algoritmo irá retornar a média e o veredito,“true” para uma boa descrição ou “false’ para uma má descrição.

A estrutura de Screw é constituída por um conjunto de módulos distintos, distribuídos pelos dois processos principais, avaliação e reparação. O sistema baseia-se num Web service, permitindo interoperabilidade para ser usado não só pelo Qualweb, mas também por outras ferramentas. Além desta característica, a modularidade foi outro aspeto relevante no desenvolvimento do sistema, evitando dependências entre módulos e facilitando os desenvolvimentos sobre este. O algoritmo apresentado é distribuído pelos módulos da avaliação: Processador de Recuperação de Informações Semânticas (SIRP), Gestor de Domínio (DM) e Inspetor de Relações (RI). O SIRP é responsável por colecionar informações resumidas sobre o conteúdo, isto é, através de interpretadores e sumarizadores é fornecido um conjunto de conceitos que representam o conteúdo em palavras, no caso das imagens, ou versões resumidas, no caso de texto. O DM é responsável por obter a aproximação semântica entre domínios com a descrição e com os conceitos fornecidos pelo SIRP. Os domínios são uma componente importante do sistema, pois valorizam a relação entre os parâmetros avaliados, no sentido em que, se um dado conceito está rela-

cionado com um certo domínio e a descrição também, então o domínio reforça a relação semântica destes dois. O RI dá a aproximação semântica entre a descrição e os conceitos, relacionando-os também com os valores obtidos no DM. O último passo da avaliação é oferecer o resultado final por meio dos módulos anteriores. O descritor do conteúdo será positivo ou negativo de acordo com o valor obtido pelo algoritmo, caso seja maior ou menor que um determinado limite, respetivamente.

Na parte de reparação existem duas fases: a fase de obtenção de novas descrições e a fase de avaliação e comparação de valores. A primeira fase reúne uma série de frases geradas por serviços externos ao sistema (atualmente); a segunda fase, avalia cada uma das novas descrições com o módulo de avaliação do Screw e compara os valores de cada um com todos os valores existentes, até encontrar o melhor valor que seja acima do mesmo limite do algoritmo. Caso não haja nenhuma descrição cujo o valor seja positivo, é gerada uma descrição estática com os três melhores conceitos retirados do SIRP e que representam a imagem.

A operação das interpretações, sumarizações, aproximação semântica e geração de novas descrições é suportada por um conjunto de serviços externos ao sistema, nomeadamente Clarifai, Indico e Swoogle. Estes serviços não são estacionários, isto é, podem ser alterados de acordo com a necessidade do desenvolvimento, beneficiando da modularidade do sistema.

Foram realizados dois estudos neste trabalho, através de questionários online, os quais permitiram definir os melhores parâmetros do algoritmo, de forma a otimizar o seu melhor desempenho. Para além disso, estes serviram para entender a qualidade das avaliações feitas pelo sistema e também serviram para entender a qualidade das descrições de imagens atualmente na Web. Esses estudos basearam-se em avaliações humanas sobre um conjunto de imagens e os seus textos alternativos (relativo ao atributo "alt"), para comparação entre as suas classificações e os resultados do sistema. O primeiro estudo permitiu afinar o algoritmo até atingir a melhor correlação possível, sendo que o melhor caso atingiu os 0,58, o que significa que é uma associação forte. No mesmo estudo são fornecidas os cinco melhores conceitos fornecidos pelo SIRP e a conclusão é que estas palavras nem sempre representam as imagens em questão. No segundo estudo, foram avaliadas todas as descrições geradas pelo módulo de reparação, no qual revelou que as frases geradas pelo sistema são no geral insuficientes como alternativas à descrição original. Por outro lado, no contexto da Web, existem muitas situações em que não existe qualquer tipo de descrição das imagens, o que afeta a leitura efetuada pelos leitores de ecrã. Apesar do valor não ser muito positivo, este módulo consegue gerar descrições que podem ser inseridas em atributos que não existem.

Por fim, esta framework acabou por ser incluída no Qualweb, para integrar novas perspetivas de avaliação da acessibilidade Web providas de avaliações semânticas. Isto é, como foi mencionado o Qualweb só realizava avaliações sintáticas e esta integração permitiu introduzir e/ou melhorar técnicas relativas a estes problemas, como por exemplo a identificação e descrição dos "alts" nas imagens. Para além desta ferramenta, foi desen-

volvido um plugin para o Google Chrome, que através dos resultados tanto do Qualweb como do Screw, concretiza reparações às páginas Web relativas às técnicas que exigem avaliação semântica de imagens.

Palavras-chave: acessibilidade web, avaliações de acessibilidade, análise semântica, páginas web

Abstract

The Internet has continuously found its way into our everyday lives, both in a professional setting as well as in entertainment. It has become an important resource for our daily activities, from work to recreation. This means that increasingly more people are browsing the WWW. There are many types of users and some of them suffer from impairments, constraining their user experience. This leads to the pursuit of an accessible Web for all types of users. This process is aided with a set of guidelines (e.g. WCAG) established by a organization, W3C. These guidelines aside from being a useful guide for Web developers, they are also used by Web accessibility tools that evaluate Web pages in order to check issues. However most of these tools cannot resort to a page's semantics and can only make syntactic evaluations. Also, they are not capable to repairing them. Therefore, this two subjects are the main objectives covered in this study: semantic evaluation and repair for web accessibility. For this purpose a tool called Screw is presented, which performs semantic evaluations to verify the relation between Web content (text and images) and their descriptions, applying an algorithm. For the repair mechanism, it generates new descriptions when the originals are considered bad by the tool. To support this development, two studies were carried, one for the algorithm's optimization and the other one to verify the quality of Screw's assessments, after the algorithm has been adjusted. For Web accessibility, Screw is integrated in Qualweb, a Web accessibility evaluator, in order to improve its evaluations to a new stage with semantic evaluation. Additionally, a plugin for Google Chrome browser was developed to repair Web pages in real time, according to Qualweb and Screw's results.

Keywords: web accessibility, accessibility evaluation, semantic analysis, web pages

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

People are increasingly relying on the Internet. Today, about 40% of the world population browses the Web, which contrasts greatly with the meager 1% of 1995 [47]. Additionally, the concern with user experience (UX) is also growing. However, there is a wide range of different users, including people with disabilities. Those users, representing almost 15% of the world's population [49], do not interact with the world in the same way as everyone does, despite having similar needs, including browsing the Internet. Considering that most of Web apps are user-centered, this demands improvements in Web accessibility.

1.1 Context and Motivation

With such a diversity of users, making Web applications accessible for everyone is an imperative task. However, designing solutions for those impaired in some way, leads to a greater challenge. There are several disabilities and creating a unique solution for all of them is an almost impossible task. Some of them are too specific or a complex combination of special conditions so, they require a more elaborated solution. An example is SWAT [40], a mobile application to suit the particular requirements of an user with tetraplegia, residual movements, blindness and low speech capability. This is a clean case that standard solutions can not solve. In other cases, finding a solution to a problem can raise further problems for others. For example, glaucoma and macular degeneration are disabilities with opposite solutions. In the former case, the condition describes progressive loss of peripheral vision; in the latter, the loss spreads from the center outwards. Following this thought, we have to consider many aspects when we think about Web accessibility such as the target user, their conditions (age and health), context of usage, the environment, among others.

Disabled people often seek supporting tools to help them accomplish routine tasks. The Internet is no exception. Its limited accessibility demands the usage of assistive technologies (screen readers, braille displays). However we need to adapt Web pages to make this viable.

There are two main ways to test if a website is sufficiently accessible or not. Those

ways are with users and experts [2]. The former, allows to gather real contextual information. For example, while they browse and explore the Web, they can "think-aloud", providing a more thorough review. However, it's not easy to gather representatives of this users group, it can be more expensive and can take a long time to reach a complete analysis. Another option to perform evaluations on Web pages is to assess their compliance to a set of standards. Usually these tasks are assigned to experts, who can be more technical than users, since they already have the acquirement to be more specific about what is failing. Expert evaluations, conducted by a human expert can be complemented with automated analysis performed by using tools.

Regarding these evaluations, the most considered standards are from Web Accessibility Initiative (WAI) [8], a unit of the World Wide Web Consortium (W3C) [11]. The WAI provides resources to help make the Web accessible to people with disabilities. Through real Web users stories, they acknowledge and formalized a set of disability' types that usually restrict the access to the Web. They're mostly related to hearing, cognitive, physical, speech and visual disabilities, assuming that using a computer or a smartphone demands a visual, touching and sometimes hearing/speech interaction. To fulfill this purpose, the organization rearranged a set of guidelines [10, 3] with heuristics (criteria and techniques) to support Web designers. This also allows the creation of Web accessibility evaluation tools, based on this set. Nowadays, there are many applications that can perform automatic evaluations such as aChecker[1], Qualweb[2], TotalValidator[3], WAVE[4], among others, that will be discussed further. Usually the process follows these steps: retrieve the source code, perform an assessment and output it in a way these users can understand.

1.2 Scope

The evaluation tools mentioned above are focused on a syntactic level, i.e., they only verify the source code syntax. For instance, checking if some tags are present on the document, like <title>, or if the header tags (<h1>,<h2>,<h3>,...) are structured hierarchically. They don't analyze if those elements make sense in context or the presence of an image's description and its meaning according to the proper image. This is important because when a Web user uses a screen reader, it will try to read to him/her informations about the structure of the page, like headers, lists, anchors, images descriptions and much more. When the screen reader identifies the tag, it needs something that can describe that element in order to give context to the user. Those descriptions can be achieved by using metadata or even by other tag elements. The simplest example is describing images through the "alt" attribute, which frequently doesn't exist or is empty [32], translating into "blank" to the user. Even when there is a description it is necessary to check the alignment between the description and the actual image meaning. If an image is about a

[1] http://achecker.ca/checker/index.php
[2] http://qualweb.di.fc.ul.pt
[3] https://www.totalvalidator.com/
[4] http://wave.webaim.org/

car on the road, but it is described like “Lemons on the road” this will mislead the user. Assessing the accessibility of Web pages at this level implies a different approach than just looking at the elements’ syntax - it is necessary to consider also their semantics.

The utility of the contributions of this thesis will be even greater if integrated with a Web accessibility evaluator. This will be Qualweb [24]. This is one of the few known that can evaluate Web accessibility of dynamic content [22]. Dynamic content are the changes that happen in Web pages without the needs of user interaction, for example when a feed is updated with new information. Those changes represent states of the DOM (Document Object Model), usually performed by Javascript, a heavily used technology on websites. Qualweb is an application implemented by the group HCIM-LaSige at Faculdade de Ciências da Universidade de Lisboa. The author of this thesis has been part of it in order to improve this technology and designing other solutions about Web accessibility issues. This gave a background and knowledge about its functioning.

In short, there are accessibility issues that need to be rectified with semantic analysis. Additionally most of the tools ask to “verify manually”, because the process of interpreting semantics correctly can only be achieved by humans. However, the goal is to make this process as automatic as possible, putting aside the need of a human hand. Converting media data (text, images, videos and links) to an understandable human language allows to relate it to what is already written by humans intended to describe that data. Tools and techniques capable of relating the content written by humans and what is conveyed by media are thus required. There are some techniques and pratices that aim to help with this process [45], allowing developers to correct the problems. For example, HTML5 nowadays provides markups like or to emphasize a text content or define roles of elements with attributes, like role=“menuitem” which identifies an item from a menu. However, that is not enough [19], because in general Web developers don’t change their practices and there is a lack of technologies to improve this situation.

1.3 Objectives

The objectives of this work are to semantically analyse and repair Web content in order to mitigate Web accessibility issues which are described in the chapter 2. Specifically, we want to achieve the following goals:

1. Conduct a review of the state of the art of Web accessibility and Web semantics, getting to know existing solutions and characterizing the problems that can be solved through the use of semantics.

2. Implement evaluation mechanisms of Web content in order to define if the content and its descriptor is semantically related.

3. Implement automatic repair mechanisms to assist developers in writing accessible code and provide users with more accessible solutions.

Table 1.1: Plan

	Task	Duration
1	Related Work	1M
2	Preliminary Report	1.5M
3	Architecture & Algorithm's design	3M
3.1	Implementation: Evaluation Module	2M
3.2	Survey 1	1M
4	Implementation: Repair Module	0,75M
5	Survey 2	1M
6	Qualweb's Integration	0,75M
7	Plugin	1M
8	Final Report	1,5M

4. Improve the assessment of Qualweb relating semantically Web content (images, text, links) with the above mechanism.

1.4 Work Plan

In this section the work plan will be described. Table 1.1 shows the tasks that were done and their duration. Note, some tasks were done in parallel or are sub-tasks of others, as will be explained.

1.4.1 Plan Description

Related Work Before starting the development, a study of the state-of-art of Web accessibility evaluation and semantic content analysis was performed. Also, the evaluator Qualweb was studied in order to understand its functionality. Plus, a research about automatic repair systems was conducted and specific tools were identified. This task took around 1 month.

Preliminary Report This report stated the related work and some research on the tools mentioned in the Related Work chapter. Part of this report was written in between with the Related Work task.

Architecture & Algorithm's Design This task is about designing the algorithm that measures the semantic relation between two contents. This task is sub-divided in two others:

Implementation: Evaluation Module This was the most time-consuming task. Within the process of developing the evaluation module, the architeture and the algorithm were defined.

Survey 1 After implementing the evaluation module, a survey was carried to understand the quality of the current state of the system in that time and also to aid tweaking the algorithm. An online form was avaiable for 1 month.

Implementation: Repair Module Development of the second module of the system to repair Web pages.

Survey 2 Second survey was handled to understand the quality of the repair module and the evaluation's module with a new set of images. This survey was online around 1 month.

Qualweb's Integration This task was about understanding the techniques that needed a semantic evaluation and covering some of them with the integration of Screw in Qualweb.

Plugin Implementation of a plugin for Google Chrome browser, to repair the pages accordingly with Qualweb's and Screw's results. Part of the plugin was made during the first survey and was finished after integrating Screw in Qualweb

Final Report Writing of the final report, describing all the tasks and their results.

1.5 Contributions

The work reported in this document lead to the contribuitions listed below:

1. An algorithm that computes the semantic similarity between a content and its descriptor;
2. An innovative tool called Screw, that evaluates semantic relations between Web contents, through the algorithm mentioned above;
3. A repair system capable of replacing bad descriptions with new ones;
4. An analytical comparison of the evaluation of image descriptions made by the automatic evaluator and humans;
5. A plugin that repairs Web pages in real time for semantic accessibility issues detected in images;
6. Insertion of Screw engine in the Qualweb tool, turning it in one of the first Web accessibility evaluators to cover semantic Web accessibility issues.

1.6 Document Structure

The report will follow this structure:

Chapter 2 Related work This chapter presents a full analysis of the state-of-art, divided into 5 main aspects: Web Accessibility Standards, Evaluation Tools, Qualweb, Automatic Repair and Semantic Content and Web Acessibilty;

Chapter 3 Screw This chapter describes the implemented tool, Screw, with an explanation of the whole system, including the algorithm, all the modules and the first study.

Chapter 4 Integrations This chapter describes all the integrations that have been benefited from Screw, the Qualweb and plugin;

Chapter 5 Evaluation In this chapter the second survey is analysed and compared with the results from the first survey;

Chapter 6 Conclusion In this chapter there is a summary of what has been done and the conclusions about the results, plus the future work with some ideas that couldn't be accomplished during the work time.

Chapter 2

Background and Related Work

This section covers the main aspects required to understand what was already made in the fields of Web accessibility and semantic content analysis. It begins by examining the standards, providing an overview of the existing evaluation tools and presenting an in depth analysis of Qualweb (how it works). This is followed by a discussion of how both domains connect, Web accessibility and semantics. Aspects related with ontologies, data mining, automatic repair and generation of image's descriptions will be treated as well.

2.1 Web Accessibility Standards

As was introduced, the organization W3C - World Wide Web Consortium, created an unit called WAI, which stands for Web Accessibility Initiative. They created a set of strategies and guidelines for different contexts related with accessibility issues. They are:

- **Authoring tool** - Addresses software that creates websites (e.g., Content Management System or HTML editors)
- **User-agents** - Addresses assistive technologies (e.g., browser or screen reader)
- **Web Content** - Addresses information in a Web site (e.g., text or images)

Each one links to a set of suitable guidelines. Their main goal is to help designers to build better websites improving their coding practices and help all users to have a better user experience.

For the Web content, the guidelines are called Web Content Accessibility Guidelines (WCAG). Its first version, WCAG1.0, was published in 1999, but with the Web's evolution it has become an incomplete and weak setting, due to the lack of principles and techniques that were necessary for the new tecnologies [9, 39]. Nine years later, a new version was created, WCAG2.0, with more detailed and helpful information.

In this last version, four principles were established [7], which are the bases followed by the guidelines (see Table 2.1).

There are 12 main guidelines, subdivided into others. Each one has a set of success criteria built according to conformance levels [6] - A, AA and AAA. These describe

Table 2.1: Description of WCAG2.0 principles

Perceivable	Concerns about the presentation of the interface, which has to be always presentable in way that can be perceived
Operable	Additional to the first principle, all interface components have to be operable in some way
Understandable	All users must understand the information and the associated operations needed to complete their tasks
Robust	The content should be as reliable and interpretable for as many user-agents and assistive technologies as possible

different requirements for different situations, the first represents a low level of success criteria satisfaction and the AAA represents a high level. A website can satisfy all success criteria on level A without satisfying the other levels. Also, usually achieving all success criteria with AAA is difficult, but if it does it can benefit more users.

Criteria goals are achieved by meeting techniques and each one is fully described to help developers to satisfy them. The result of the assessment can be pass or fail, i.e., success or failure. Some of them are categorized as being sufficient to pass a criteria or advisory to help improve some other aspects not covered. Its description, besides its meaning, provides applicability, usage examples, related techniques and a procedure for tests. Some of these techniques require a semantic content analysis to be well succeed. These ones are described in table 2.2. For example, for h25 [1] the procedure section goes like the following:

Procedure

1. Examine the source code of the HTML or XHTML document and check that a non-empty title element appears in the head section.
2. Check that the title element describes the document

Expected Results

- Checks 1 and 2 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

[1] https://www.w3.org/TR/WCAG20-TECHS/H25.html

The first checkpoint is related to the presence of a <title>tag in the <head>section, i.e., syntax evaluation, and the last one is about checking the title's description congruence with the document, i.e., semantic evaluation. The two distinct fields are expected to be accomplished for the success of the technique,"Checks 1 and 2 are true".

This technique can satisfy a success criteria from the guideline 2.4, related with the navigation and location of the user in a website. The success criteria 2.4.2 says that a Web page must have a title which describes it.

> **Guideline 2.4 Navigable**: Provide ways to help users navigate, find content, and determine where they are.
>
> **2.4.2 Page Titled**: Web pages have titles that describe topic or purpose. (**Level A**)

2.1.1 WAI-ARIA

WAI-ARIA [4] (Accessible Rich Internet Applications) is a specification of WCAG with the purpose to improve the accessibility of dynamic content. As said, Web is evolving and nowadays content can be dynamic with the introduction of new technologies such as AJAX (Asynchronous Javascript And XML). Hereupon, these guidelines discriminate roles (with a taxonomy) that can be assigned to HTML elements in order to provide semantic information that can be used to enhance accessibility. Those roles define the behavior of that element considering its context. For example, an item inside a list would get an attribute *role="listitem"*. Additionally to roles, the guidelines define states and properties. These provide information about an object. The difference between both is that properties are immutable and states change due to user interaction. This gives structural information to assistive technologies that can rearrange it into an easier format for the user to understand and browse, like, for example, using the role *scope* to identify columns and rows in tables or items in a list.

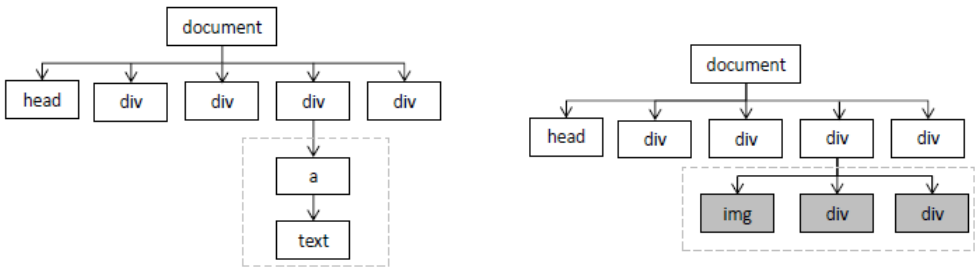
2.1.2 Post-processing and States

The source code of a Web page is a document that can be represented with a hierarchical structure called DOM, Document Object Model, which mainly is a structured tree with all the elements as objects.

The term post-processing is related with the loading process of the DOM by the browser [24]. First, Web page's resources are requested and parsed to build the DOM tree of the Web page. The resources are ready to be used by the user when the DOM is fully loaded. Wihtin this loading there are some events that can be triggered before the user gets a chance to interact. This behaviour is often associated with dynamic Web pages. Post-processing is the time when the DOM is completed loaded with all the initial events triggered in the Web page.

Table 2.2: WCAG techniques with semantics

Techs	Description
h2	Combining adjacent image and text links for the same resource
h25	Providing a title using the title element
h30	Providing link text that describes the purpose of a link for anchor elements
h33	Supplementing link text with the title attribute
h36	Using alt attributes on images used as submit buttons
h37	Using alt attributes on img elements
h39	Using caption elements to associate data table captions with data tables
h42	Using h1-h6 to identify headings
h43	Using id and headers attributes to associate data cells with header cells in data tables
h44	Using label elements to associate text labels with form controls
h45	Using longdesc
h48	Using ol, ul and dl for lists or groups of links
h49	Using semantic markup to mark emphasized or special text
h53	Using the body of the object element
h54	Using the dfn element to identify the defining instance of a word
h57	Using language attributes on the html element
h58	Using language attributes to identify changes in the human language
h64	Using the title attribute of the frame and iframeelements
h65	Using the title attribute to identify form controls when the label element cannot be used
h69	Providing heading elements at the beginning of each section of content
h70	Using frame elements to group blocks of repeated material
h73	Using the summary attribute of the table element to give an overview of data tables
h75	Ensuring that Web pages are well-formed
h77	Identifying the purpose of a link using link text combined with its enclosing list item
h78	Identifying the purpose of a link using link text combined with its enclosing paragraph
h79	Identifying the purpose of a link in a data table using the link text combined with its enclosing table cell and associated table header cells
h80	Identifying the purpose of a link using link text combined with the preceding heading element
h81	Identifying the purpose of a link in a nested list using link text combined with the parent list item under which the list is nested
h83	Using the target attribute to open a new window on user request and indicating this in link text
h85	Using OPTGROUP to group OPTION elements inside a SELECT
h89	Using the title attribute to provide context-sensitive help
h90	Indicating required form controls using label or legend
h97	Grouping related links using the nav element



(a) Third div with an anchor element

(b) Third div with three other elements (img and two divs)

Figure 2.1: Changed states from (a) to (b), from some triggered action (dynamic content) [22]

A state is when a change in the page occurs. For example, clicking in a button to login and then pop-ups the form, the DOM is temporarily affected what may lead to new Web accessibility issues. This type of triggered events usually are made through Javascript technologies, like AJAX. This allows the content to be dynamic, like displaying/hiding information (as the previous login example), injecting new content (posting on social network) and even removing content (deleting a post). Fernandes et al [24] assumed that each change of the DOM represents a new state and they found a way to detect those states, through a Web crawler capable of triggering the unloaded events. They also concluded that there are differences between regular and post-processing evaluations, which hightlights the value of Qualweb's assessment.

There is an example on figure 2.1 about states, illustrating what happens to DOM when some event is triggered. The first figure shows the DOM of a set of a document's objects, where the third div has an anchor and a text as its children, but when an event happens, for example clicking in the anchor, the DOM changes and its children became one image and two divs.

2.1.3 Are WCAG2.0 enough?

There are a few studies which evaluate the efficiency of WCAG2.0 standards. For example, Sayago [44] tested a specific guideline (2.4) related with links, on older people. They created a set of expressions used on hypertext. Among other conclusions, they found that using "click here to..." improves accessibility when embedded in paragraphs, but is not recommended by the guidelines. Despite the specific type of audience, they could conclude that the examples given by WCAG are ambiguous and lack information on how this criteria should be applied. This study was done in 2009 and the WCAG instruction pages could have been modified since then.

However, a more recent study [42], shows that disabled people had around twice more problems than normal users. For blind (full or partial) people, most of the problems were related to links. They were confused when using similar or same words on both text and links; or the amount of links in the same page was too high, causing text-to-speech to

become annoying and harder to understand. For motor impaired users, the most common problems were related to the size of the elements and the bad linkage (i.e., a link only on the button's text). Alongside the detection of these issues, they compared the efficiency of WCAG1.0 and 2.0, and concluded that the first version only detected 27% of the problems reported by users and the second just 32%. An expectable result, as version 2.0 inherits guidelines from 1.0, where guideline 2.4 is one of them, therefore there will not be a big difference between both evaluations.

Broader studies like [38, 26] also conclude that WCAG is not sufficient to guarantee Web accessibility and it can not be the only resource to accomplish it. This is an important verdict, because it shows the lack of coverage of some aspects.

2.2 Evaluation Tools

As was introduced before, we can evaluate Web pages with automatic evaluation tools. Table 2.3 lists some of these tools, but there are many more besides these ones. Every tool in the table was tested and explored to collect its properties.

The tools were evaluated according to the following features:

- **Post-processing**, an important feature since there are millions of websites built using Javascript[5] and granting them the ability to use dynamic content. Javascript can change the state of websites without loading a new URL, by triggered scripts. This can influence the accessibility of the page, because content that was not present in the source can be rendered to the user.

- **Upload File** allows the developers to check short snippets instead of an online Web page.

- **Conformance Levels Filter** is a feature related to the quality of the output, so that it could be filtered by the levels - A, AA or AAA.

- **Shows Code** is important because it contextualizes the location of the assessments for each technique, in the code.

- **WCAG Versions** gives some perspective about the version used to make assessments, since version 1.0 is it outdated compared to 2.0.

All of these technical features are important, but the most relevant for this thesis are **Code Repair** and **Semantic Analysis**. **Code Repair** is related to automatic repair of the code in any level (syntactic or semantic) and **Semantic Analysis** is about performing assessments semantically.

As shown in the table no tool can perform **Semantic Analysis**, with most of them asking for manual checks. This important result reflects the general state of Web accessibility evaluations. Also, we can see that only three tools have Post-processing: Google's

Table 2.3: Comparison of automatic accessibility evaluation tools

Tools	*Post-processing*	*Upload File*	*Conformance Levels Filter*	*Shows Code*	*WCAG Versions*	*Code Repair*	*Semantic Analysis*	Observations
AChecker[1]	No	Yes	Yes	Yes	Both	No	No	Use other guidelines besides WCAG. Evaluation of pieces of code.
Total Validator[2]	No	Yes	Yes	Yes	Both	No	No	Only desktop application.
TAWdis[3]	No	No	Yes	Yes*	Both	No	No	*Unstructured and messy. Low detail
Access Monitor[4]	No	Yes	No	Yes	Both	No	No	Gives a rating of accessibility. Only in Portuguese.
Accessibility Developer Tools[5b]	Yes	No	No	Yes	N/I	No	No	Plugin only for Chrome. Own sheet code for issues.
A-Tester[6]	No	No	No	Yes	2.0	No	No	Weak visual presentation, no filters only continuous text
WAVE[7]	Yes	Yes	Yes	Yes	Both	No	No	Show location error on the page
Qualweb[8]	Yes	No	No	Yes	2.0	No	No	Gives a percentage of Web accessibility

a - IP ; b - Google ; c - WebAIM ; N/I - No infomation. Created at November 28, 2016

[1] http://achecker.ca/checker/index.php
[2] https://www.totalvalidator.com/
[3] http://www.tawdis.net/
[4] http://www.acessibilidade.gov.pt/accessmonitor
[5] can be downloaded on Chrome Web Store
[6] http://www.evaluera.co.uk/
[7] http://wave.webaim.org/
[8] http://qualweb.di.fc.ul.pt

tool, WAVE and Qualweb. For the **Repair Code** none of them has a working mechanism that could be seen, so this thesis is innovative in this regard also. Regarding **Upload File**, **Conformance Levels Filter**, **Shows Code** and **WCAG Version** it's possible to see that there are some variations, most of these tools still gives the possibility to evaluate with version 1.0.

The last tool reviewed, Qualweb, lacks this type of analysis too. Since this is the tool that will benefit from the work described in this document, the next section describes in detail how it works.
In sum, with this analysis we can conclude that none of the evaluators are capable of semantically assessing Web pages, which leads to one of our objectives to analyse semantic Web content in order to improve Web accessibility.

2.3 Qualweb

Similiar to other tools, Qualweb obtains the source code of the page, then evaluates across the techniques and outputs the results to the Web browser. Qualweb [22] is a two-sided application, it's possible to do evaluations on both server and client-side, the first in the command-line and the second through browser (web service or website). For each assessment Qualweb gives a Web accessibility percentage, related to the number of passed tests and outputs the evaluation results in three possible ways: a CSV file for statistic purposes, a JSON Object that can be filtered to be presented on a website; and in EARL format, an accessible standard format.

2.3.1 Architecture

Qualweb's server is composed by three main components: the core, browser processor simulator and interaction simulator. This architecture is illustrated in the figure 2.2.

The Core

When an entry on the Qualweb's website is made, the request is sent to the server with the related URI **(step 1)**. Upon reception, the server forwards the URI to the Browser Processing Simulator module (BPS) to be processed and evaluated **(step 2)**. If there is more than one state of the page, the DOM is sent to the Interaction Simulator (IS) **(steps 3 and 4, these steps can be repeated)**. After simulating the DOM states and being processed by the BPS and IS, the processed DOM is sent back to the core **(step 5)**. Towards the extraction of the CSS, Qualweb uses a CSS pre-processor, that reaches both code on the current page and in the CSS files **(step 6)**.

After extracting everything possible to create the raw DOM, this item will be parsed with the package HTMLPARSER, filtering into a JSON format easier to make callbacks. Then the DOM is evaluated through the techniques' scripts **(step 7)**, generating the various outputs **(step 8)** and sending the resulting JSON to the browser **(step 9)** formatted in

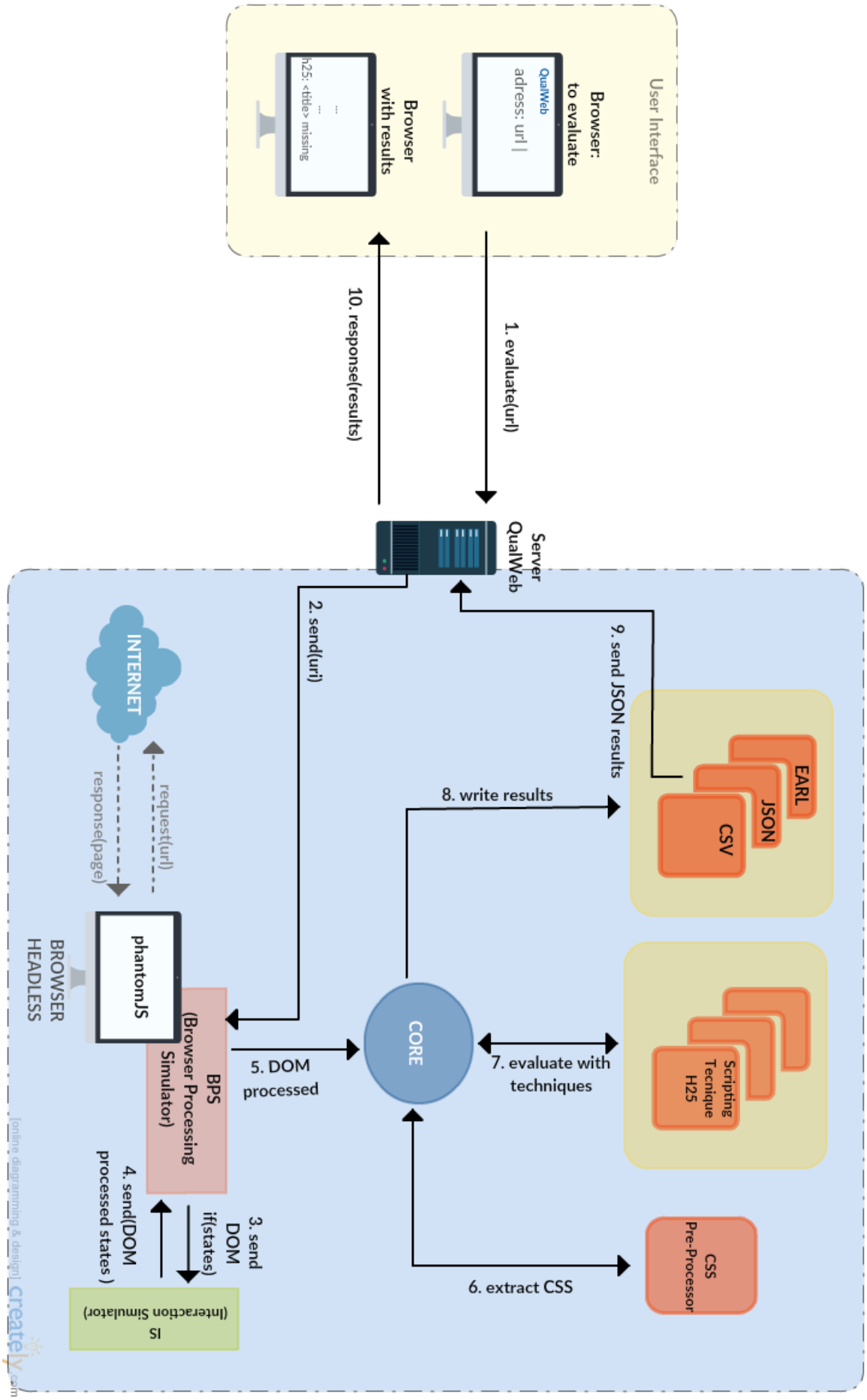


Figure 2.2: Qualweb's architecture

a readable output **(step 10)**.

Browser Processing Simulator

This module uses the PhantomJS to simulate browser requests. PhantomJS is a headless browser, in other words, without a graphical interface and with a CLI (command-line interface), which uses Webkit that allows for fast rendering of Web pages. All these conditions enable quick and large-scale multi-processing.

Interaction Simulator

This component is responsible to simulate the states of the Web page and then get them as a DOM result. For each state the child states are forked and each one is sent to the BPS. At the end there is an interaction graph of all states. As was said to achieve this type of evaluation, a Web crawler is used to execute all possible trigger events.

2.3.2 Techniques

Overall, Qualweb implements 47 HTML and CSS techniques. Table 2.4 presents all the success criteria and related techniques used in Qualweb.

2.4 Automatic Repair

As shown in the table 2.3 there is a lack of automatic repair features in Web accessibility evaluators and the only act of repair they have is through suggestions. When we seek for a technique on their output, besides the description, some of them give repair suggestions of how to do it. Although, there is AccVerify/Repair [2] an evaluator that can perform repair, but its usage is restricted to the users from their institution, which means there is no access to retrieve some useful information about this tool.

W3C provide a document [18] where, for each technique, they describe its current state, the evaluation (elements and requirements), suggested messages, ways to repair and test files. This can be a huge contribution for evaluation applications' developers.

On the other hand there are a few technologies that can be found in [35] which can perform some type of automated repair. Mainly their focus is about buggy software at code level. Since accessibility issues are not considered a bug, it deviates from this thesis scope. However, it is closely related and there are some of them working on web's field, like Carzaniga et al [17] which collects the failures of a Web page into a report and then changes the code according to a set of rewriting rules, designed by the authors. Those rules have properties that will define the way how they're applied.

Tidy [3] is only a repair tool for HTML markup, which has a page discussing about Web accessibility but does not perform this type of evaluations. ARROW [48] is another

[2] http://warc.calpoly.edu/accessibility/accverify.html
[3] http://www.html-tidy.org/

Table 2.4: Qualweb techniques

Success Criteria	Techniques (H - HTML, C - CSS)
1.1.1	H2 H24 H30 H35 H36 H37 H44 H45 H46 H53 H65 H67
1.2.3	H53
1.2.8	H46 H53
1.3.1	H39 H44 H51 H63 H71 H73 C22
1.3.2	C6 C8
1.4.1	C15
1.4.4	C12 C13 C14 C17 C20,C28 C22
1.4.5	C12,C13,C14,C8,C6 C22
1.4.8	C12,C13,C14,C8,C6 C19 C20, C24 C21 C23 C25
1.4.9	C12,C13,C14,C8,C6 C22
2.1.3	H91
2.1.3	H91
2.2.4	H2
2.4.1	H64 C6
2.4.2	H25
2.4.4	H24 H30 H33 C7
2.4.5	H59
2.4.7	C15
2.4.9	H2 H24 H30 H33 C7
3.1.1	H57
3.1.6	H62
3.2.2	H32 H84
3.2.5	H76
3.3.2	H44
3.3.5	H89
4.1.1	H93 H94
4.1.2	H44 H64 H91

application that makes automatic repair of Web pages, but different from the previous tool. This one tries to resolve race issues. In this context, race is a concept to asynchronous problems on functions callbacks on client-side. This occurs many times on Javascript scripts, for example having a function A and B and there's something on B that needs A to loads first, but happens exactly the inverse creating failures. They detect all races precedents (creating the DOM based on invocations), create a causal graph of them, each node represents a type (function, html element, among others) and edges the precedent's type of the nodes. Another tool is Facil'iti [4] which creates a kind of overlay that changes the presentation of the website according to a specific user profile, but still isn't a Web accessibility evaluator.

All of these tools don't solve the problems that we are trying to resolve. One of our objectives is to repair code in order to improve Web accessibility and none of them is capable of it.

2.4.1 Transcoding

There is a term called transcoding which is a "general concept of transforming content or a program on the fly in an intermediary server, resulting in other formats" and transcoding for Web accessibility "is a category of techologies to transform inacessible Web content into accessible content on the fly" [14]. Most of the tecnologies that make this procedure usually use anotation-based or rule-based techniques which influence the CSS display, like changing colors, schemas and text sizes. They also rely on domain specific languages (DSL), to create new annotations or to apply pages' segmentation techniques to analise their structure, like VIPS and HearSay. An example of these annotation-based techniques can be seen in this study [13], where the authors try to transcode Web pages to improve their accessibility for blind people.

The most used methods for transcoding are: Text Magnification, Color Schema Changes, Serialization, Alternative Text Insertion, Page Rearrangement and Simplification. The first two are related with the design of the page, already meantioned in this section, such as text color and sizes. The Serialization, Page Rearrangement and Simplification are associated to the layout of the page, where some elements are removed if they are unnecessary, changing the page's layout to be readable by screen readers.

The Alternative Text Insertion is the one related with this work, since one of the goals is to analyse images and generate descriptions (alternative texts) when necessary. It's admited that "We still lack any tecnology that can enter appropriate alternative texts for each image" [14]. This paper is from 2008, so nowadays we can find some technology to aid this thread, however they are not used for Web accessibility purposes yet.

Qualweb [23, 21] has a mechanism for automatic repair based on templates, but currently is not working online. Also, as a Web server has a great condition to perform transcoding being an intermediary server for browser evaluations. However, the end users

[4] https://www.facil-iti.com

are not capable to see the evaluations since the output it's only a full report of the techniques which are failing and passing according to the Web page's DOM. So a new approach is needed to get those evaluations to be seen by users.

2.5 Semantic Content and Web Accessibility

First of all, what does "semantic" means? It is the study of the meaning of words.

In the world of science this subject is applied to a structure that can be accomplished with ontologies [15, 27] (also considered a graph of concepts). They are very present in Biology and Medicine fields, artificial intelligence, semantic Web and information architecture. Ontologies have instances, classes (types of objects), attributes (properties of an instance) and relations (between classes and instances) to represent information. To represent this structure on the Web, XML, RDF or OWL are the most used. For example, dbpedia [5] is a semantic Web resource for many domains, like football, which can be used through proper languages like Turtle or SPARQL. These two are a query language for RDF schema, which allows to get linked data.

2.5.1 Relations and Ontologies

For the assessment of semantic content it is necessary to verify the level of relationships between words, like synonymy, antonymy, homonymy, hyperonymy/hyponymy and others. Some accessibility issues require the identification of the meaning of Web content, like images, videos, even paragraphs, in order for screen readers to transmit the information to the users correctly without failures. The paradigm behind this relies on: the closer the relationship between the content and its description, the higher the probability of having a more accurate description. For example, every dictionary is a general ontology. There are Web applications that their ontology is published and can be somehow used to retrieve words' relations.

Wordnet [6] has its ontology online, that can be used with some API's[7,8]. Its technology works like a dictionary, however they offer more than just definitions, they provide relations between words, like synonyms and meronyms, types like domains and instances (ISA statements) and much more. For each word the API returns a synset with some of the properties mencionted above. A synset is a definition for a set of synonyms, each expressing a distinct concept. As a full result, each synset can be a node to reach other words, which means a more detailed tree inception. Figure 2.3 is an example of a synset for the word *Amazon*, which shows one definition and an instance *river*, which can lead to another synset (marked as "S:"). More information about Wordnet can be found in [33, 34, 1].

[5] http://wiki.dbpedia.org/
[6] http://wordnetWeb.princeton.edu/perl/Webwn
[7] https://github.com/NaturalNode/natural
[8] http://www.nltk.org/howto/wordnet.html

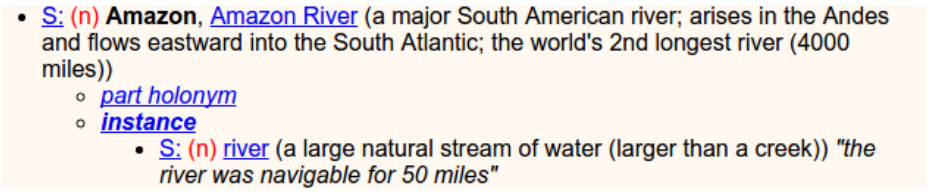


Figure 2.3: Synset example

However, Wordnet only offers searches with only one single word, which means it can not find if any set of words (at least 2) are related and how. Trying to relate two words can be a huge effort, since there are many ways of being related and no way to know their relation level in the ontology graph, in other words it can be found in one iteration or a finite large number of it. To accomplish this purpose, Swoogle - SimService [9] [25] from the University of Maryland, Baltimore County, performs an evaluation of the relation between two expressions and gives a value of their similarity. In contrast to Wordnet, Swoogle does not say what kind of relation it is.

2.5.2 Web Content Mining

For the detection of relations it is necessary to retrieve information about the Web content. Considering that the Web content is mostly based on image, text, link and video, these will be the main focus. There are tools which perform content analysis from text and image. As for links, mostly they represent a connection to a new Web site or page or event content like images/videos. Therefore, it will be mixed between text and image mining. They'll be described next.

Text

Text mining includes many types of analysis. One is highlighting the most relevant words in a text. Usually it discards words like "the" or "a" since they don't give any important information. Text mining can be achieved by tf-idf [12] algorithms, which extract the most important words from a document through weight (numbers of appearances) and type of word (noun, verb, adjective, others).

Other type of analysis is similarity of words. These words can be in the same family of words, like "mother" and "motherhood', or from the same semantic domain, like "eagle" and "owl" from the "birds" domain.

A set of tools are described:

Indico.io[10] This tool does a wide variety of analysis: sentiment, text tags, keywords, thematic (people, organizations, others), relevance, among others. Also has a free plan with 10k calls per month. Has a well structured documentation and API.

[9] http://swoogle.umbc.edu/SimService/
[10] https://indico.io/

Meaning Cloud[11] Has less variety than Indico, but with free account they allow 40k per month. Also, has an API and a fine documentation.
Rosette[12] Equals to Indico with 10k free calls per month and has a feature for relationships between words, however its documentation is a little more confusing.
Cortical.io[13] All free, but less features and a poor output compared with others.

Image and Video

Image mining is the extraction of knowledge and data through images analysis [52]. Image mining [52, 16] has to treat some issues like: the dominant color, since colors can represent objects, emotions or others things, for example the color "blue" can be related with water, sky or cold feelings and "red" can be related with danger, blood or fire, there are lots of meanings for colors and their tones [50] that can be relevant for image interpretation; object recognition is another thread to retrieve the main objects of an image allowing to identify if there are persons, animals, cars, buildings, among others present in the pictures; the presence of patterns (walls, floors, animals, colors) can also give some spatial information and contextual information of the image. Those properties will help to generate proper labels.

Cloud Vision [14] This tool is from Google, unfortunately is a trial version and wasn't tested.
Computer Vision [15] From Microsoft, only 5k free calls per month. Less result tags than Clarifai, however it provides the most dominant color as an output (an unique feature compared to others)
Clarifai [16] Equals to Computer Vision with 5k per month, in a free account. Besides image, it is the only one analyzing videos. Also has a well structured documentation and API.
Imagga [17] Only 2k free calls per month, less than the others. Allows to choose the language of the results.

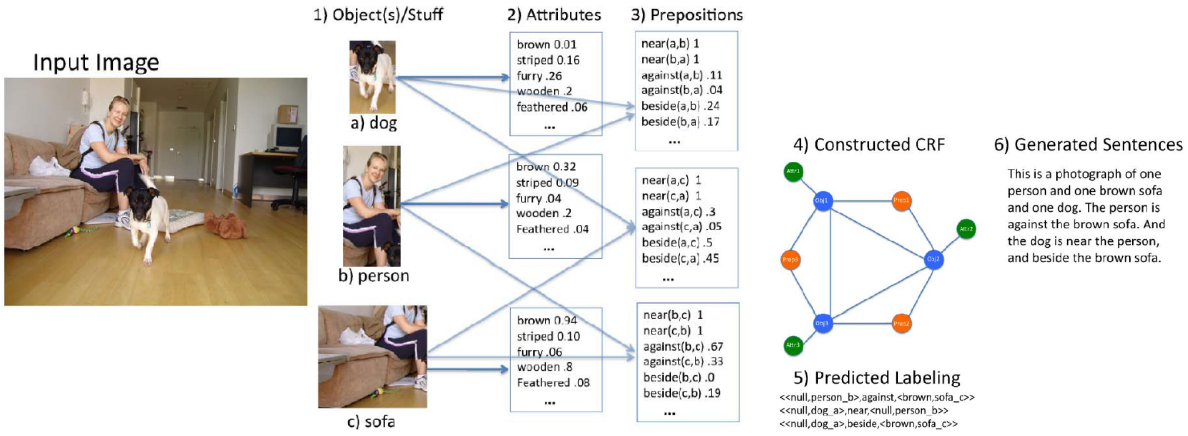
Named Entity Recognition

Data mining is also concern about the presence of domains in the content. This type of information can tell us more about their context and, therefore, a way to relate information

[11] https://www.meaningcloud.com/
[12] https://www.rosette.com/
[13] http://www.cortical.io/
[14] https://cloud.google.com/
[15] https://www.microsoft.com/cognitive-services/en-us/computer-vision-api
[16] https://developer.clarifai.com/
[17] http://imagga.com

between different sources. One pratice is through Named Entity Recognition. If the concepts are recognized as particulars words or expressions (names of entities) like nouns of persons, companies, cities, dates, among others, they can be part of domains such as persons, places, organizations, time and others [36] which can correlate.

2.5.3 Verifying and Generating Descriptions



(a) Retrieving information from the picture (b) Generation of the sentence

Figure 2.4: Description's generation from BabyTalk

In the Web there are some situations where lack of descriptions causes accessibility issues. For example, the image element should have an alternative text to give image context to blind people, but this is not often complied. So, there are two types of situations in Web, an image can already have a description or doesn't have it at all.

Those situations are already studied in the field of natural language, trying to create ways to generate proper descriptions from scratch or through inference with already known images and descriptions - machine learning.

For the latter there are many studies like Karpathy, et al. [28] and Richard Socher, et al. [46] where they use a recursive neural network model based on dependency trees for matching and aligning images with descriptions. This approach comes from learning images with similar captions in order to be capable to associate future assessments with the closest ones. In the same context, other studies [51, 31, 20] focused on similarity between images and text. The first study [51] creates a corpus of images of everyday activities to construct a large scale visual denotation graph, which associates image descriptions with their denotations. They call a denotation as a set of images that can self-describe them. Some of their work can be seen here [18]. The second work [31] demonstrates a semantic relatedness between words and images from ImageNet. They use visual codewords and textual words (like synsets of WordNet) to construct a joint semantic vector space of a given set of images in a node. The semantic vector space is a representation of concepts

[18] http://shannon.cs.illinois.edu/DenotationGraph/graph/index500.html

in a mathematical space. This allows to do information retrieval, disambiguation and document segmentation. The last study [20] is similar to the previous one, however they investigate the impact of those semantic relations between two images in the context of ImageNet.

However, these studies rely on the premise of existent descriptions, thus generating descriptions from scratch it's a whole different approach. The applications mentioned in section 2.5.2 mostly supports only single word or sometimes a small expression of two words. These form a weak caption for an image, because a single word can't fully describe it. Babytalk [30] is a research that tries to address this problem, creating a complete description without requiring related text or similar images. They predict labels in different regions of an image, which usually represent an object (like person or car). For each object they classify it using a set of trained attributes (like brown or small) based on the object's domain. Fig 2.4 shows an example of their work.

2.6 Summary

This chapter provides an overview of Web Acessibility Standards and what has been done on this field to accomplish semantic assessments.

Besides WCAG techniques, WAI introduces WAI-ARIA focused on dynamic content, an important feature of building Websites nowadays. Also the efficiency of WCAG standards is questioned, which many studies points to an insufficient source to guarantee Web accessibility of a page. A range of Web accessibility evaluators are listed and compared, concluding that none of them supports code repair or semantic analysis. Qualweb is also introduced, describing in detail its architecture and its outstanding feature of post-processing evaluation. Some tools are discussed in order to perform automatic repair. In semantic content, the notion of relationship between words is explored as well as image description generation and ways to retrieve information from Web content.

While there is isolated applications that can perform evaluations or repair code, in sum there is no evaluator that can perform both simultaneously. Additionally none of the Web accessibility tools repairs code, even less when is Web semantic content.

Therefore this work will forward towards this matter, resolving issues of Web accessibility by the presence of a better semantic content.

Chapter 3

Screw

In this chapter, the characterization of the semantic accessibility evaluator will be presented. Initially, its architecture, structure and functioning are introduced with a detailed description of each module related with the implemented algorithm. Then, a survey that was held during the development process will be thoroughly analysed, which provided results to adjust all the algorithm parameters. Lastly, the adjusted algorithm is defined.

3.1 Architecture and Design

Screw was created as a web service making it a more undependable and interoperable tool. Besides the interoperable aspect, modularity was also desired. Modularity creates an abstraction of all code and allows to introduce more modules without affecting any other code that is independent of the new module. This benefits further additions of any appropriate module, since the tool is not fully implemented and third-party services may change.

Figure 3.1 illustrates the architecture, which is divided in two main components, one that does evaluations and the other repairs, each one composed by a set of modules. In the initial step, every request is sent to the Module Decider which delivers the request's content to the main module, the core. Then the respective operation (evaluation or repair) is executed and the answer is sent back to the Module Decider that delivers it to the Web service and then to the client. The yellow boxes attached to the modules are some of the third-party services used, which will be explained after the Web Service and the Module Decider layers.

3.1.1 Webservice

As was said, Screw works like a Webservice, receiving HTTP requests everytime a semantic analysis is needed. Table 3.1 describes this service and how to perform the requests, followed by an example.

The parameters are a little peculiar because there were some constraints while developping the request. The POST body is divided by"//contentText//", where the first part

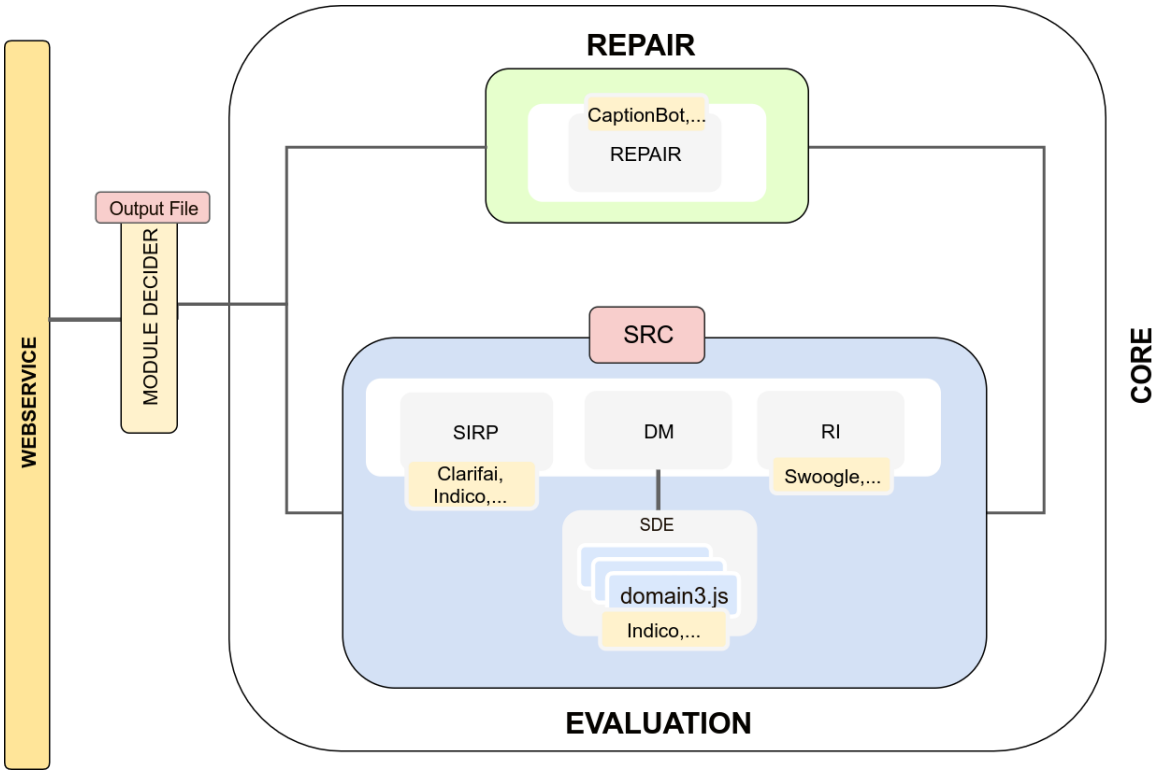


Figure 3.1: Screw's architecture

Table 3.1: Webservice API

Resource	Type	Parameters	Description
/screw/	*POST*	{"type": :type , "desc": :desc}//contentText//:param	Executes an evaluation between a description *:desc* and a content *:param*
/screw/repair	*POST*	{"type": :type , "desc": :desc}//contentText//:param	Executes a repair for the content *:param*

includes the parameters in JSON and the second a parameter with a URL or a text. In some situations, the latter can be all the content text of a page, which might have a great size and special characters that don't work with JSON structure. Even using decoding functions didn't help. The parameter *:type* is the type of the upcoming content, "1" for URL (an image) or "2" for text. The parameter *:desc* is the short text that is trying to describe the content. The result of the first request is the algorithm's output: a value, representing the semantic relation degree between the content and the description, and the verdict, a boolean value, where true stands for a good description and false for a bad one. The result of the second request is a new sentence to replace the original description.

Figure 4.10 is used as an example for a request, its URL is `http://www.telegraph.co.uk/content/dam/news/2016/12/16/MAYJS116266571_PA_Brexit-large_trans_NvBQzQNjv4BqZgEkZX3M936N5BQK4Va8RQJ6Ra64K3tAxfZq0dvIBJw.jpg` and the alternative text is *Theresa May delivers a major speech on Brexit today*. For this example, the server ran with all the 20 words to get the real time for the completed assessment, thus the final result will be different from the algorithm's output example. The webservice is running in localhost in the port 3000, so the host is "http://localhost:3000' ' . Each request has to be with POST type and the *Content-Type* header field has to be set for "text/plain". Listings 3.1 and 3.2 show an example of requests for the evaluation and repair modules and their responses.

```
Request:POST /screw/ HTTP/1.1
Host: localhost:3000
Content-Type: text/plain
Cache-Control: no-cache
Postman-Token: 9b39b1c5-3c79-b39c-a0ba-cc22a3fa6425

{
  ''type'':''1'',
  ''desc'':''Theresa May delivers a major speech on Brexit
    today''
}//contentText//http://www.telegraph.co.uk/content/dam/
news/2016/12/16/MAYJS116266571_PA_Brexit-large_trans_
NvBQzQNjv4BqZgEkZX3M936N5BQK4Va8RQJ6Ra64K3tAxfZq0d
vIBJw.jpg

Response:0.2208953:true
```

Listing 3.1: Request call to evaluation module

```
Request:POST /screw/repair HTTP/1.1
Host: localhost:3000
Content-Type: text/plain
Cache-Control: no-cache
Postman-Token: c8b59d59-04f8-3dfd-78de-755807aed8e5

{
  ''type'':''-r'',
  ''desc'':''Theresa May delivers a major speech on Brexit
    today''
}//contentText//http://www.telegraph.co.uk/content/dam/
news/2016/12/16/MAYJS116266571_PA_Brexit-large_trans_
NvBQzQNjv4BqZgEkZX3M936N5BQK4Va8RQJ6Ra64K3tAxfZq0d
vIBJw.jpg

Response:Theresa May who is smiling and looking at the
   camera, and she seems to have a neutral face.
```

Listing 3.2: Request call to repair module

3.1.2 Module Decider

The Module Decider is responsible for redirecting the evaluation to the modules that are being requested. Therefore, if the assessment is between images and text or two texts this module will run the proper module. Besides, this module has the capability to save the results to an output file with a desired format. This feature allows to run any further analysis that might be needed.

3.1.3 Third-party Services

To achieve some functionalities, third-party solutions were considered. These functionalities are related with extraction of semantic information from Web content - Data Mining. The usage of third-party services reduces the implementation's effort and allows to focus only on the main functionality of this system: generation of the semantic relation degree between a content and its description.

Consequently, one of the challenges was to find tools capable to do some mining work. Like was listed in 2.5.2, there are some services that can perform this type of analysis. Some optimal conditions to comply should be:

- No cost
- API for Javascript

Usually one of the constraints of being free is having a limited amount of API calls that can be requested to the service. The other condition is also ideal because Screw is developed in Node.js (Javascript) which means a more compatible system that can be well infused, however other languages are not discarded at all.

For information mining and abiding to these conditions there are Clarifai and Indico. As was mentioned, Indico is an application that can execute text mining and Clarifai image/video mining. Both API's documentation is fine and very helpful to understand, also they have support for Node.js. Swoogle is used to evaluate the semantic relation between two expressions. Finally, for the repair module, CaptionBot is used to create new descriptions.

Based on the modularity of this system, any module mentioned above can be replaced by any other with the same type of analysis.

Indico

Indico is a service that extracts information from texts through machine-learning models in a cloud-hosted environment. To get Indico working is necessary to install the package "indico.io" with npm and to create an account in their system to generate a necessary API Key.

Indico offers lots of functions for many aspects like: keywords, relevance, text tags, language predictor, persons, places, among others. Those functions work with Promisses (a feature of Node.js). Each one outputs the assessment result and the respective confidence degree, between 1 and 0.

Listing 3.3 is an example of an Indico's function. In line 20th the wanted function, *keywords*, is being called and will output the most relevant words from a text, *txt*. The results are being filtered by the `indico_res` function.

Say that one statement of `txt` is "I love to code, but I need coffee", the output will be like:

(key: "coffee", value: 0.2700564069 ,
key: "love", value: 0.24602579460000001 ,
key: "code", value: 0.2415576506 ,
key: "need", value: 0.1632670523)

Usage in SCREW This service is mainly used to make specific assessments of descriptions. As Indico has the capability of retrieving specific domains, like persons and places, the results can be used to determine the relation of the description to certain domains. This procedure will be explain in 3.2.2.

```
1  var indico = require(''indico.io'');
2  indico.apiKey =  //API KEY
3
4  //function called by indico.
5  // It filters the result
6  function indico_res(res) {
7    for( var key in res ){
8      tmp_indicoKey.push({
9        concept: key,
10       confidence: res[key]
11     });
12   }
13
14   tmp_indicoKey = tmp_indicoKey.sort(sortBy(''-confidence
       ''));
15   tmp_indicoKey = tmp_indicoKey.slice(0,20)
16
17   callback(null,tmp_indicoKey);
18 }
19
20 indico.keywords(txt)
21   .then(indico_res)
22   .catch(logError);
```

Listing 3.3: Indico's script in Screw

Clarifai

Clarifai is a service that extracts information from a image's URL through recognition models. To get Clarifai working is necessary to install the package "clarifai" with npm and to create an account in their system to generate a necessary API Key.

Clarifai offers some models: general (*GENERAL_ MODEL*), food (*FOOD_ MODEL*), logotypes (*LOGOS_ MODEL*), faces *FACES_ MODEL*), among others. These models represent domains, which redirect the prediction to specific concepts. Those functions work with Promisses (a feature of Node.js). Each one outputs the assessment result and the respective confidence degree, between 0 and 1.

The *predict* function makes a prediction of the possible tags, as can be seen in listing 3.4. The result is a JSON array of the 20 most relevant tags over an image URL, *imageurl*.

Using fig. 3.2 like an example, the filtered result should be like:

(concept: 'no person', value: 0.97437674 ,
concept: 'little', value: 0.96982837 ,
concept: 'cute', value: 0.962057 ,
concept: 'domestic', value: 0.9416512 ,
concept: 'mammal', value: 0.9328555 ,

concept: ‘pet’, value: 0.928562 ,
concept: ‘nature’, value: 0.9260636 ,
... +13)

```
1   //function called by clarifai.
2   // It filters the result
3   function clarifai_res(response) {
4
5       var allconcepts = response[''outputs''][0][''data
            ''][''concepts''];
6
7       for( var c in allconcepts ){
8         var concept = allconcepts[c]
9         tmp_clarifai.push({
10          concept: concept[''name''],
11          confidence: concept[''value'']
12        });
13      }
14
15      tmp_clarifai = tmp_clarifai.sort(sortBy(''-confidence
            ''));
16    cb1(null,tmp_clarifai);
17  }
18
19  app.models
20    .predict(Clarifai.GENERAL_MODEL, imageurl)
21    .then(clarifai_res);
22 }
```

Listing 3.4: Carifai’s example script

Figure 3.2: Clarifai example

Usage in SCREW Currently this module is only using `GENERAL_MODEL`. In time of development this was the only one available, however increasing the number of models could give richer assessments. This is used in the module Semantic Information Retrieval Processor, explained in section 3.2.1.

Swoogle

As was said in 2.5.1, Swoogle is an evaluator that gives a value of similarity between two given expressions. It has three types of analysis: top-n, phrase noun/verb and STS. The first one gives the top-n most similiar words to an input word, the second gives the semantic similarity between nouns or verbs and the last one the semantic textual similarity between two phrases.

To ask for an evaluation simply do a HTTP request. For example:

```
http://swoogle.umbc.edu/SimService/GetSimilarity?
operation=api&phrase1=There%20was%20a%20car%
20crash%20on%20the%20road&phrase2=The%20traffic%
20was%20slow%20because%20of%20an%20accident
```

Output: 0.5013031

In this request we can specify the corpus and the type of the evaluation. The corpus can be from *Refined Stanford WebBase* (Webbase) or *LDC English Gigawords Corpus* (gigawords). The type can be *concept* or *relation*. They can be added do the URL with "`&corpus=:corpus&type=:type`".

Usage in SCREW In SCREW, the Swoogle module is used to get semantic similarity between a description and the concecpts given by Clarifai or other module with similar purpose. Currently it performs evaluations with phrase verb/noun, corpus=*Webbase* (the default) and type=*relation* and *concept*. The module that uses Swoogle will be explained in section 3.2.3.

CaptionBot

CaptionBot is a robot that is capable of creating descriptions for images, developed by Microsoft.

For this service, there isn't any official API only some helpful packages created by the community.

For Node.js there is a npm package available called "CaptionBot", which can be used as demonstrated in listing 3.5 where *imageurl* is the parameter for the image url that is being described.

```
1 CaptionBot(imageurl)
2     .then(caption => {
3         console.log(caption
              );
4     })
```

Listing 3.5: CaptionBot's example script

Usage in SCREW At the time of development, the Node.js package wasn't working. For this reason, we were forced to call this servive through a Python script, which is available as well. This is only used in the Repair mechanism to create descriptions of images. This integration is explained in the section 3.3.

3.2 Evaluation Module

The evaluation module is capable of doing two types of assessment. One that checks the relation between an image and a text and the other the relation between two texts. After the reviewing process upon the WCAG2.0 in chapter 2, these were the most relevant cases found for when a semantic assessment is needed.

The evaluation's process runs accordingly to an algorithm. The algorithm's objective is to calculate the semantic similarity between a content and its description, i.e., compute if the description is describing the content. Since description should refer the most important concepts of the content, and the referred content can be rather complex, we first aim to obtain the most relevant concepts associated with the content. If these concepts are highly related with the description, consequently the description has a high probability to describe the content.

This content can be a text or an image, although the algorithm has been tested so far with the latter only. A set of values measuring the semantic similarity between the contents and its description can be gathered, evaluating relations between them and domains through third-party services. The domains are a component of the algorithm that improves the relations of the content and the description. Based on the results of this work, the usage of domains were considered relevant to calculate the value of the relation similarity between the description and the content. Specifically, if the content is associated to a set of domains and the description is associated with them as well, then both have a major probability to be related semantically.

Therefore, the algorithm works upon a list of concepts, a description and a set of available domains. The algorithm follows this steps:

- Calculates the semantic relation between each concept and the description
- Calculates for each domain:
  - The semantic relation with the description
  - The semantic relation with the each concept
  - The semantic relation between each concept and description in the domain
- Calculates the value of the semantic relation between the description and the content using the previous calculated relations

The last calculation will give the semantic similarity between the description and the content. All these algorithm's steps will be detailed in further sections.

In figure 3.1 you can see how the modules are connected. The core receives the parameters from Module Decider and runs orderly all the modules to evaluate and collect the necessary data to calculate the similarity between the given description and the image or text. These modules will be explained in the next sections. The output examples are from the image example used in the Algorithm (4.2) section.

Inside the core the order goes like the following table 3.2. The first column is the step number of the running function in the evaluation flow, comprised by 6 steps. The column "Module" is the acronym of the respective module:

1. SIRP - Semantic Information Retrieval Processor
2. DM - Domains Manager
3. RI - Relations Inspector

The third column is the main function of the module that is being called in the core's side. The last two columns are the inputs and outputs of each function. Essencially, every main function feeds a JSON structure that can be passed to other functions through temporary variables, but in the end everything will be aggregate in a vector with all the JSON results. This object, besides every result of each module, will also include the final result, i.e., the similarity value obtained by the Semantic Rating Calculus.

In technical terms, taking advantage of Node.js and its easiness to work with asynchronous code, most of the implementation is handled with async functions of Async library. In listing 3.6 there is a representation of the core's construction with an async.series() function. This code runs every function in the respective order and each result will be stacked in the final JSON object, *results*, until the series is over. When it's over, the results are sent back to the Module Decider through the callback call.

Table 3.2: Evaluation's functions

Step	Module	Function	Input	Output
1	SIRP	clari	imageurl	concepts
2	DM	checkConcepts	description concepts	concepts_domains
3	DM	checkDescriptionDomains	description	desc_spec_domains
4	DM	addDescValueByDomain	description desc_spec _domains	desc_domains
5	RI	checkSwoogle	description concepts concepts_domain desc_domains	rel_swoogle
6	(in core)	calcRelation	rel_swoogle	final_result

```
1  exports.initImageCaption = function(imageurl, description,
       cb){
2
3    async.series([
4        function(callback) { SIRP.clari... },
5        function(callback) { DM.checkConceptsDomains... },
6        function(callback) { DM.checkDescriptionDomains...
           },
7        function(callback) { DM.addDescValueByDomain... },
8        function(callback) { RI.checkSwoogle... },
9        function(callback) { calcRelation... }
10   ], function(err, results){
11     cb(null, results);
12   });
13 };
```

Listing 3.6: Evaluation's main function

3.2.1 Semantic Information Retrieval Processor

The Semantic Information Retrieval Processor (SIRP) is the first one to be called by the core. This call executes a sub-module providing a set of concepts that somehow describe and are related with the content. When the content is an image, the concepts are provided by the Clarifai sub-module. The necessary input will be only the image, *imageurl* and the output is a JSON structure with two keys, *concept* and *confidence*. Listing 3.7 is an example of an output.

$c = (concept, confidence)$ a concept from Clarifai, where *concept* is the concept's designation and *confidence* is the confidence degree of the relation between $concept$ and the image

C a set of all concepts describing an image

```
C = 1 [
    2   {
    3       ''concept'': ''market
                ''',
    4       ''confidence'':
                0.9948592
    5   },
    6   ...+19
    7 ]
```

Listing 3.7: SIRP output

3.2.2 Domains Manager

While developing the algorithm it was found some relevance in taking the usage of domains as an important feature of the process. The use of these domains provides a more accurate evaluation over the system inputs, a URL image and a description, and hence allows to discriminate the contexts of both inputs. If the description and one of the concepts of the image are related with the same domain, we can consider that the concept and the description are related too, like if A is related to B and C is related to B, probably A is also related to C. This will strengthen the semantic relation between the description and the image.

$D_{\mathbf{n}}$ a set of domains with size n

d a domain, where $d \in D$

Currently we use only two domains:

$$D = \{persons, places\}$$

In this module there are two main relations to be considered for the domain's evaluation: the relation between domains and concept and the relation between domains and description. The former is collected through the Swoogle's service, where for each domain and each concept a relation value is achieved. The output for the example, figure 4.10 is in the listing 3.8.

$SCDom$ a set of relations between all concepts and all domains

$ScDom = (D, c, values)$ a relation between all domains D and a concept c, where in *values* there is the result for the each respective domain, $c \in C$

$Scd = (d, c, value)$ a relation between a domain d and a concept c, where *value* is the result, $d \in D$ and $c \in C$

$SCDom =$

```
[
    {
        ''domain'': ''persons'',
        ''concept'': ''portrait'',
        ''value'': 0.048137933
    },
    {
        ''domain'': ''persons'',
        ''concept'': ''one'',
        ''value'': 0
    },
    ...+9
    {
        ''domain'': ''places'',
        ''concept'': ''portrait'',
        ''value'': 0.045044
    },
    {
        ''domain'': ''places'',
        ''concept'': ''one'',
        ''value'': 0
    },
    ... +9
]
```

Listing 3.8: Relation between concepts and domains output

For the relation between the domains and the description there are two approaches: a **general evaluation** and a **specific evaluation**. Each result of each evaluation will be considered for the relation value between both parameters.

The **general evaluation** is achieved through Swoogle's service as well and there will be a single value for each relation between each domain and description. Using the example of the figure 4.10 with the description *Theresa May delivers a major speech on Brexit today*, the values are:

persons: 0.050429244
places: 0.03381388

The **specific evaluation** will extract specific words from the description related with each existing domain (*persons* and *places*) through Indico's service. This process is called Named Entity Recognition as was introduced in Background and Related Work chapter. For the domain *persons* a set of persons' names present in the description will be collected, as well for the *places* domain with locations' names. In the code this domains are executed in a certain order, first is *persons* and then is *places*. The example from figure 4.10 with the listing 3.9 shows that Indico detects the name "Theresa May"(an entity) with a confidence degree of 0.67 ($SDde$, where d is *persons*) and no location's names were found (with an empty array in the output).

GDD (General Description Domain) a set of relations between the description and all domains of the general evaluation

$GDd = (description, d, value)$ a relation between a domain d and the description of the general evaluation, where *value* is the result, $d \in D$

$SDDE$ a set of confidence degrees of entities found in the description in all domains, where *value* is the result

$SDde = (description, d, entity, value)$ the confidence degree of an entity found in the description in the domain d, where *value* is the result, $d \in D$

SDD (Specific Description Domain) a set of relations between the description and all domains of the specific evaluation

$SDd = (description, d, value)$ a relation between a domain d and the description of the specific evaluation, where *value* is the result, $d \in D$

$SDDE =$

```
[
    [
      {
        ''person'': ''Theresa May'',
        ''confidence'': 0.6699638367
      }
    ],
    []
 ]
```

Listing 3.9: Specific evaluation output with all the entities found in the description

$$SDd = S\bar{D}de$$

Within this process, multiple entities($SDDE$) can be found for each domain, therefore to get a single value an average for each domain is calculated with the confidence degree of each entity found(SDd). In this example, since there is only one entity, its confidence degree, 0.67, remains as the single value of the relation between the description and the *persons* domain and for the *places* domain, with no entity found, the single value will be 0.

After determining the values from the general and specific evaluations, a calculus is made to get the final result for each relation with the domains and the description. All the values are between 0 and 1, therefore the final result will be given by the general evaluation's value and the specific evaluation's value weighted by the difference between 1 and the general evaluation's value. The specific evaluation value is only used to increase the general value in order to enhance the relation between the description and each domain.

The final result is between 0 and 1, as well.

FDD (Final Description Domain) a set of final results between the description and all domains

$FDd = (description, d, value)$ a final result between a domain d and the description, where *value* is the result, $d \in D$

$$FDd = GDd + (1 - GDd) * SDd$$

where $d \in D$

With the example, corresponding to the values in table 4.4 from the Algorithm's section, the calculus is as below with the output being like the listing 3.10:

persons: $0.050429244 + (1 - 0.050429244) * 0.6699638367 = 0.68660731$
places: $0.03381388 + (1 - 0.03381388) * 0 = 0.03381388$

$FDD =$

```
[
    {
        ''domain'': ''persons'',
        ''value'': 0.68660731
    },
    {
        ''domain'': ''places'',
        ''value'': 0.03381388
    }
]
```

Listing 3.10: Final result output of Domains Manager

New Domains Insertion

Presently, the system only has two domains. Allowing the insertion of more domains might give better results. Thanks to the modularity of the system, introducing more domains is easier.

There are two places where the domain must be inserted. The first place goes with the module itself, which might be developed using a service to retrieve entities, therefore it should be placed in the path */lib/third-party*. After being developed it has to be imported to the DM and inside the *checkDescriptionDomains* function there is an async function, similar to what is in the core, and the new module should be introduced here. Listing 3.11 present the current state of the function and an example of a new module, *organizations*.

```
1  exports.checkDescriptionDomains = function(description,
       callback){
2
3    async.series([
4      function(cb){
5        indico.persons(description,cb);
6      },
7      function(cb){
8        indico.places(description,cb);
9      },
10     function(cb){
11       indico.organizations(description,cb);
12     }
13   ], function(err,results){
14       callback(null, results);
15   });
16
17 }
```

Listing 3.11: checkDescriptionDomains function

The second place is related with the average calculated with the entities' confidence degree. It's mandatory to create a sub-module which executes calculations to retrieve the single value, as was explained previously. This module's file should have the same name that is in the vector and should be implemented inside the directory */lib/domains/*. Also, in this sub-module, the main function has to be exported as *"init"*, since it will be called equally like the others.

Note, to insert these new modules they must have a way to retrieve entities, in accordance with the new domains.

3.2.3 Relations Inspector

The previous sections showed how to gather information about the concepts (representing the image), the semantic relation between domains and concepts, and between

the domains and the description. In this module the relation between the concepts and the description is computed, using Swoogle service.

$SCDesc$	a set of relations between the concepts and the description.
$ScDesc = (description, c, values)$	a relation between a concept c and the description.
$SCDD$	a set of relations between the concepts, the description through semantic relations with domains.
$ScDd = (description, domain, c, d, value)$	a relation between a concept c_i and the description, where *value* is the result, $c \in C$ and $d \in D$

After retrieving those relations semantically, we have now two values for each concept: the value between the concept and each domain(Scd) and the value between the concept and the description($ScDesc$). Reminding, if the description and a concept have some value in the same domain, the concept will earn the best value of both. Therefore, a maximum formula is applied to each concept between these two values. This formula was defined through the study reported in section 4.1. In the end, there will be a vector of all the values after this calculation, like in the listing 3.12.

$$ScDd = max[ScDesc, Scd]$$

where $c \in C$ and $d \in D$

$SCDD =$

```
[
    {
      ''concept'': ''
         portrait'',
      ''rel_value_rel'':
         0.048137933
    },
    {
      ''concept'': ''one'',
      ''rel_value_rel'': 0
    },
  ...+8
]
```

Listing 3.12: RI output

3.2.4 Semantic Rating Calculus

The last step of the process is to calculate the final result that tells if the description is really describing the image or not. It's difficult to define whenever a description is good or bad, since usually this type of analysis is made with human judgement. In section 4.1 will be explained a study where we try to understand how close these calculations are from people's judgement in order to achieve the best automatic result. Our approach has only 2 classes, describes and not describes, which is defined through a threshold, which was calculated in the optimization process described in the same study. After evaluating everything, the result will be provided by the output given by the RI, which has the values for each concept of the image related with the description.

The algorithm does an average of all concepts' values whose value is not null. The calculated value is considered bad or good if it is lower or higher than a defined threshold. Also, the number of zero values in the set of the relation between concepts and description is counted, if the number of zeros exceed a percentage, also calculated in section 4.1, the algorithm considers the description a bad descriptor for the image.

The threshold and the percentage values were defined in the study in section 4.1.

$$avgAllconcepts = \frac{\sum_{c \in s_{>0}} ScDd[value]}{s^{=}_{>0}},$$

$$s^{=}_{>0} = \{ScDd \text{ where } value > 0\}, c \in C \text{ and } d \in D$$

The last module's output will be the *avg* value and a boolean which is false when it's a bad descriptor or true when it's a good descriptor.

$$final\_result = {}^{1}\,|\,\text{``0.2644003:true''}$$

Listing 3.13: Final result output

As was said the final ouput will be an aggretion of every module's output.

$$[[C], [SCDom], [SDD], [FDD], [SCDD], final\_result]$$

3.2.5 Textual Content

The algorithm mentioned above was described for images, but the system is also capable of perfoming evaluations in a textual context. Even though the process is similar to both types, there are some details that must be explained.

The difference to be discussed is about the way how the words are retrieved, in the first step (table 3.2). Instead of using Clarifai, a text summarizer is used from Indico, which gives a set of the most relevant words in a text. Apart from using a summarizer, Indico also provides a topic retriever which was previously tested, but considered pointless for this purpose, because it's limited to its own set of words. Another issue to consider is what type of characters the content has, because probably might have some useless

symbols, like punctuation, which can misrepresent and twist the result of the summarizer. Therefore, before sending a request to the Webservice the string must be cleaned, i.e, the punctuation, the delimiters, must be removed. To help further developments over this tool, some suggestions are presented:

- If the requester is developed in Nodejs, there are two npm packages to help decode Web pages' text content: *iconv-lite*, to convert the HTML content into a readable UTF-8 string (this is important because of the existence of special characters in some languages) and *textversionjs*, to gather only the present text in the retrieved HTML, i.e. ignores the HTML tags.
- To remove the symbols, use a regular expression (`^a-zA-Z\u00C0-\u024F s|`) to split the text. This will give a vector with all the words except the unnecessary characters.

3.3 Repair Module

The second process mentioned in the beginning of this chapter will be discussed here. In contrast to the evaluation, which supports two types, this system currently only repairs if the content is an image. Despite not being implemented, the ideas to accomplish a repair for text content will be discussed later in the section.

This repair system is independent of the evaluation process, although the result of the evaluation will decide if the assessed content needs to be repaired or not. So, if the final result is lower than 0.14 (*Threshold*) the content will be repaired with the intent of providing a new description.

The structure of this development is analogous to the evaluation but with only one module with two main functions, seen in table 3.3 and listing 3.14.

Table 3.3: Repair's functions

Step	Module	Function	Input	Output
1	Repair	getCaptions	imageurl description	captions
2	Repair	evaluateCaptions	captions imageurl description	final_caption

The algorithm begins by generating new descriptors with third-party services. Currently, for this purpose, CaptionBot, which was described in section 3.1.3, is the only module being used. As was said previously, this module is executed as a Python script due to some issues with the npm package avaiable. To do so, the Python script imports the CaptionBot Python package, receives the image's url, runs the CaptionBot and the

result is sent to the main function. On the *repair* module side, after receiving the sentence from CaptionBot, the sentence is treated in order to reduce the sensation of automation, molding into a more natural expression as possible. Specifically, the sentences provided by CaptionBot, start with "I think it's a..." and this is always removed before sending to the core.

In the second function, after collecting all the new possible descriptors in the *getCaption* function, each one is evaluated by Screw. Each result is compared to find the best value. Using the same *Threshold* (0.14), if none of the descriptors is equal or higher than it, all the descriptions are discarded and a fixed sentence is formalised with the three concepts obtained by the Clarifai module that better represent the image. This standard sentence is: "This image is mainly about these concepts: <1st best concept>, <2nd best concept>and <3rd best concept>".

```
1 exports.initRepairImageCaption = function(imageurl,
    description, cb){
2
3   async.series([
4       function(callback) { repair.getCaptions... },
5       function(callback) { repair.evaluateCaptions... }
6   ], function(err, results){
7     cb(null, results);
8   });
9 };
```

Listing 3.14: Repair's main function

3.3.1 Additional Developments

One problem about CaptionBot is that it sometimes puts emojis at the end of sentences. This situation can raise some issues for screen readers while trying to read out loud those sentences. Assuming that not all of the assistive tools are capable of encoding these characters, a converter was created.

This module is located inside the directory */lib* and its name is *convertSmiletoText*. First, finding a pattern was necessary to reduce the amount of possibilities and, as was mentioned, we found that they only put smilies at the end of the sentence. To ease this translation there is a table with all the emojis and respective information: description, native icon associated, bytes and r-encoding. This table is in the file *emDict.csv*. Another pattern found was that almost every one of them starts with the bytes `\xF0` or `\xE2` and it's always 4 bytes or 3 bytes respectively. So every time there is an occurencce of one of these types, the next 4 or 3 bytes are pulled and transformed in a string, allowing a comparison with the strings inside the table. Next, if this set of bytes is part of the table, the emoji description is retrieved and reconstructed without many textual gaps. In listing 3.15 there is an example of a conversion with the sentence "I want to eat  " (the sentence in the example is not given by the CaptionBot).

```
1 var convertemoji = require('emoji-totext');
2 var str = ''I want to eat [pizza emoji]'';
3
4 convertemoji.toText(str,function(err,sentence){
5     console.log(sentence);
6  })
7  //Output: 'I want to eat a slice of pizza‘
```

Listing 3.15: Emoji converter example

3.3.2 New Repair Modules Insertion

To insert new modules for the creation of more images descriptions, the function *getCaptions* needs to be modified. Similar to the Domains Manager's process, all that is required is to export and include inside the *async.series* a new function with the module's call.

3.3.3 Textual Content

Currently, the repair mechanism doesn't support text repair, so a future development will be needed to complete the basics of the system. The biggest problem to achieve this is to find a tool capable of retrieving a meaningful sentence of a full text, instead of a summarizer. Without this it's very hard to generate appropriate descriptions. The process should be the same as the image repair, however there are more constraints with text analysis than image analysis that must be considered, as the language in which is written, the special characters, text layout (if it's relevant) and others.

3.4 Summary

In this chapter, a new tool called Screw was presented, which performs evaluations and repairs semantically Web pages' content. More specifically, its algorithm allows to evaluate if descriptors really describe their attached content, like images. Its architecture was described, as well each module that composes the system and the algorithm.

Chapter 4

Evaluation Module Algorithm

4.1 Parameters Optimization

The algorithm went through an optimization process before reaching the final version. This version is in the pseudocode below. By this snippet there are five parameters that influences the algorithm's result: $Threshold_{SRC}$, *zeroRatio*, *Formula*, *Threshold* and *Swoogle's type*.

1. $Threshold_{SRC}$ - This parameter is related with a measured threshold to calculate the semantic relation value between the concepts and the description with only a set of best words.
2. *zeroRatio* - This parameter is related with the ratio of zeros in the vector of semantic relation values between the concepts and the description.
3. *Swoogle's type* - This parameter is related with the type of query made to Swoogle, to calculate semantic relations between two expressions.
4. *Formula* - This parameter is related with the formula used to calculate the semantic relation value between the concepts and the description through semantic relations with domains.
5. *Threshold* - This parameter defines if the description describes or not the content, through the final value.

To optimize the parameters a survey was held during the development. Besides, with the survey analysis we were able to answer to some questions, in order to retrieve useful data related with the functioning of the system and its components, and also about the quality of the alternative texts provided by websites. These were:

1. Are the resulting values of the system related to the values rated by people?
2. Do different domains impact differently the classification results?
3. Are the words offered by the system considered good descriptors by the people?
4. Are the classification order of words of Clarifai related to the classification order rated by the people?

Algorithm 1 Algorithm to be adjusted

Input: $concepts, description, domains$

1: **for** $domain$ in $domains$ **do**
2: GDD [$domain$] ← Calculate general SR ($description$, $domain$)
3: SDD [$domain$] ← Calculate specific SR ($description$, $domain$)
4: FDD [$domain$] ← Calculate final SR($description$, $domain$,$SDD[domain]$,$GDD[domain]$)
5: **end for**
6: **for** $concept$ in $concepts$ **do**
7: SCDesc[$concept$] ← Calculate SR ($concept$, $description$)
8: **for** $domain$ in $domains$ **do**
9: SCDom[$concept$,$domain$] ← Calculate SR ($concept$, $domain$)
10: **if** Exist some value in FDD[$domain$] **then**
11: SCDD[$concept, domain$] ← Calculate **FORMULA** (SCDom[$concept$,$domain$], SCDesc[$concept$], FDD[$concept$,$domain$])
12: **end if**
13: **end for**
14: **end for**
15: $avgAllconcepts$ ← Calculate average of all SCDD values > 0
16: $zRatio$ ← Calculate ratio of zeros in SCDD
17: T ← Calculate $avgAllconcepts*$ **THRESHOLD$_{\mathbf{SRC}}$**
18: **if** $zRatio >=$ **ZERORATIO then**
19: $finalValue \leftarrow 0$
20: **else**
21: $finalValue$ ← Calculate average of all SCDD values above T
22: **end if**
23: **if** $finalValue <$ **THRESHOLD then**
24: $verdict$ ← false
25: **else**
26: $verdict$ ← true
27: **end if**

Output: $finalValue$ and $verdict$

5. Do people classify the original alternative texts as good descriptions?

The following sections will discuss the parameters to optimize, the setup and then the results and their analysis.

4.1.1 Parameters

The parameter *Threshold$_{SRC}$* is used on the last step of the whole process, Semantic Rating Calculus. The first approach to build this algorithm was to compute an average of words but considering only the ones above a threshold (*T* in the pseudocode, line 15). This *T* value is determined with the average of all words, *avg*, multiplied by the parameter *Threshold$_{SRC}$*. The ideia behind this, was to use only the best words of the whole set of concepts (of each evaluation). While implementing this approach, we observed the values from Swoogle were inconstant, so we couldn't use a fixed value but one that should fluctuate according with those values. This means, some words had low ratings compared with other image's evaluation, although in the range of words for the same image they were still the best values and good qualifiers. The final result will be TAVG (*TAVG*) value, which is the average between all the words above the threshold that is calculated dinamically using *Threshold$_{SRC}$*. This was tested with values parameter is between 0 and 1, with intervals of one tenth.

$$AVG = \frac{\sum_{c \in s_{>0}} ScDd[value]}{s^{=}_{>0}}$$

$$T = AVG * \text{Threshold}_{\text{SRC}}$$

$$TAVG = \frac{\sum_{c \in s_{>T}} ScDd[value]}{s^{=}_{>T}}$$

$$s^{=}_{>0} = \{\text{total of } SCDD \text{ where } value > 0\},$$

$$s^{=}_{>T} = \{\text{total of } SCDD \text{ where } value > T\},$$

$$c \in C \text{ and } d \in D$$

The second parameter is the percentage related with the ammount of zero values in the set of concepts, when related with the description. The motivation is, if there are more zero's than positive values, according to the parameter's value, this means that there are a set of words that don't describe the image at all and so, the final result should be considered false. This parameter, *zeroRatio*, can be between 0 and 1.

$$zRatio = \frac{\sum_{c \in s_{==0}} ScDd[value]}{s^{=}_{=0}},$$

$$s^{=}_{=0} = \{\text{total of } SCDD \text{ where } value = 0\},$$

$$c \in C \text{ and } d \in D$$

In section 3.1.1 was mentioned that Swoogle has at least two types of assessments: *concept* and *relation*. Another value is also considered, the average between both types. Therefore, the third parameter *Swoogle's Type* is related with this setting, having three ways, *relation*, which is the default parameter for Swoogle, *concept* and an average between the results of *relation* and *concept*.

The fourth parameter is related to the calculus made in section 3.2.3, where the *value* is set using the maximum of the value of the relation of the concepts with the domains and relation's value of the concepts with the description. Before adopting this approach, there was another calculus where one tried to consider both values. With this formula, both values contribute to the result. Consequently, this parameter will have two options, maxmium formula or the calculus with both values:

$$ScDd = \frac{(ScDesc + (1 - ScDesc) * Scd) + (Scd + (1 - Scd) * ScDesc)}{2}$$

$$\text{where } c \in C \text{ and } d \in D$$

Finally, the last parameter is the threshold used in SRC to conclude if a description is good or not for the image being assessed. Note that it's a different value from the first threshold that was previously defined. The former gets the best words in a set of concepts and the last one defines the verdict of the evaluation.

4.1.2 Survey Setup

In this survey, thirty different images were collected, together with their own alternative text. All of these alternative texts needed to be in English, because the system only treats expressions in this language, and also they should come from real contexts. Therefore, these images were mainly collected from The Telegraph and The Sun, since they're journal Websites and probably often used by all types of users. The images were chosen following a criteria of diversity, with persons, animals, buildings, landscapes, objects, among others. In the appendix A there is a table with all the images with the respective alternative text.

The survey was scripted with Google Form's language and was composed by two parts (A and B). Part A's goal is to understand how well humans rate the images' alternative texts and how their judgment compares to the system's evaluation. For the second part, instead of rating a description, is offered a set of words, given by Clarifai in SIRP's module. The objective is to acknowedge if those words are related or not with the image based on people's opinion.

The form was shared via e-mail (mailing lists of institutions and universities) and social networks. We know we would reach many people who don't have English as their

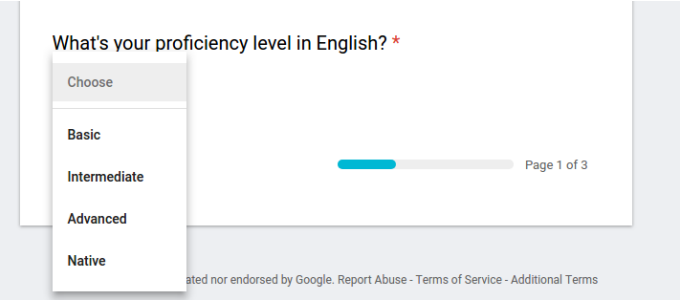


Figure 4.1: Survey's English proffiency level

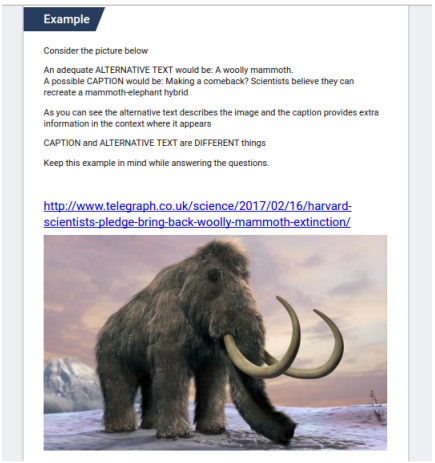
native language and for that reason we asked about their Enligsh profficiency level, offering four levels (Basic, Intermediate, Advanced and Native) as is shown in figure 4.1. No other demographic questions were considered important to ask, due to their irrelevance to the study.

Before starting the first part, the difference between captions and alternative texts was clarified to participants, assuming that people are not familiar with the meaning of both concepts within the accessibility context. Despite the call to this difference in the introduction, there is an example (figure 4.2a) in the beginning of part A with an image and two sentences, one as a caption and another as an alternative text. Part A provides an image and its alternative text and asks to rate how well this sentence describes the image in a scale of 1 to 4, where 1 is the worst case (figure 4.2b). In part B (figure 4.2c) is presented the best five words of each image, obtained by the SIRP module (Clarifai service). Similar to part A we ask to rate between 1 and 4, how well each concept describes the image. The purpose is to understand if the words are good descriptors and if the classification order is the same in both evaluations, by the users and Clarifai.

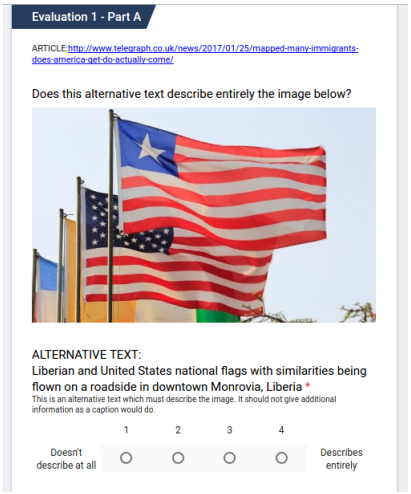
To aid this study a database was designed to avoid the amount of HTTP requests made to third-party services, making the system evaluations quicker, since the system relies on this calls to operate. Figure 4.3 present the UML schema. This optimization is important because we needed to execute the system mutiple times when trying different parameter values, with the same inputs. All the values of the process are kept in the database except for the final result. Three tables save the known information, like the alternative texts, images, domains and the concepts. The concepts are important, since they come from the same source and sometimes evaluations have the same words in their set. The other three tables are the results of every relation result. One additional advantage is that this allows to reproduce a variety of results with the different parameters without having compromised values, because of possible changes in the third-party services.

4.1.3 Results

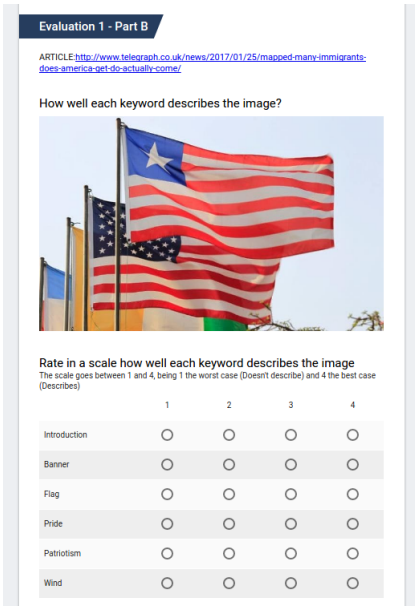
As a result, 192 responses were collected from distinct people. The data was analysed with R scripts. Starting with the proficiency level, most people answered as being Advanced, with 42%, and Intermediate, with 22%, moreover the Native were 34%, as can be



(a) Aid post



(b) Survey's part A



(c) Survey's part B

Figure 4.2: First survey sample

seen in figure 4.4.

For the parameters optimization a R script was implemented to execute the algorithm 726 times, with all the possible combinations. Table in the appendix B shows the value of each parameter already explained in the previous sections and the correlation between the algorithm and the survey answers. Note that at this stage, the system was setted up with the *persons* domain only as a way to optimize the other parameters. The correlation used was Spearman's correlation, as the results given by people follow a distribution that is not normal, with the Shapiro-Wilk test resulting in a $p - value \simeq 0$, which rejects normality and the usage of Pearson's correlation.

After gathering all the data, 5 configurations, in bold, achieved the maximum correlation value of 0.51. It was considered that $Threshold_{SRC}$ parameter isn't relevant for the

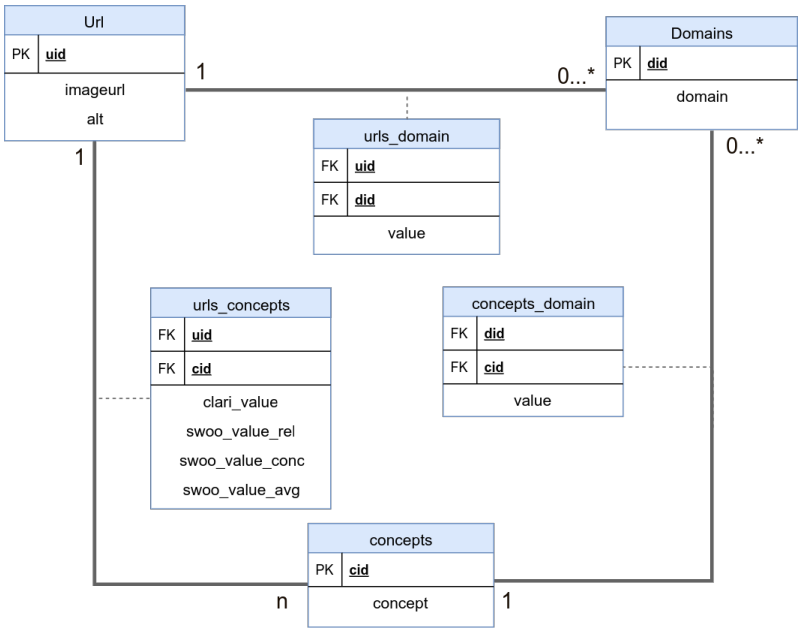


Figure 4.3: Survey's UML schema of the database

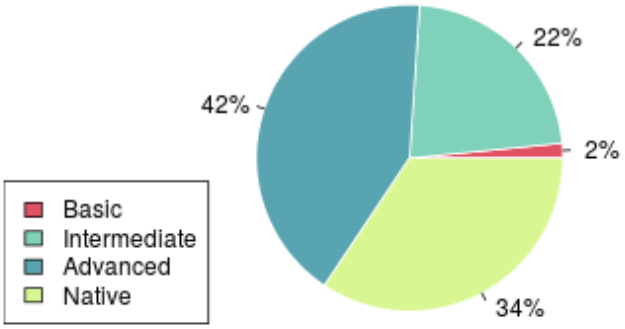


Figure 4.4: Survey's proffiency level results

algorithm, since is always zero in all maximum configurations. For the *zeroRatio*, any value above 0.5 can be used, which means that for every case where the number of zeros is superior to the number of positive values, the description can be treated as bad for the image. The best way to request Swoogle similarities is with type *concept* and to get the best value for each concept the maximum formula is the one to be used, instead of the measure with the weight of each value. This configuration is the one that was described during the charaterization of the whole algorithm.

Another interesting result is the pattern found in the evaluations, as can be seen in the plot shown in figure 4.5. Based on an analysis over this pattern, each occurrence of it represents the values of $Threshold_{SRC}$, 0 to 1, and an analysis of it shows when the *zeroRatio* is higher than 0.5, the correlation has a higher value too and the negative values of the plot represents the configuration when the *zeroRatio* is zero. For the last values of threshold, there is a progressive decrease of the correlations.

Figure 4.6, presents another analysis, where each image represents a set of evaluations

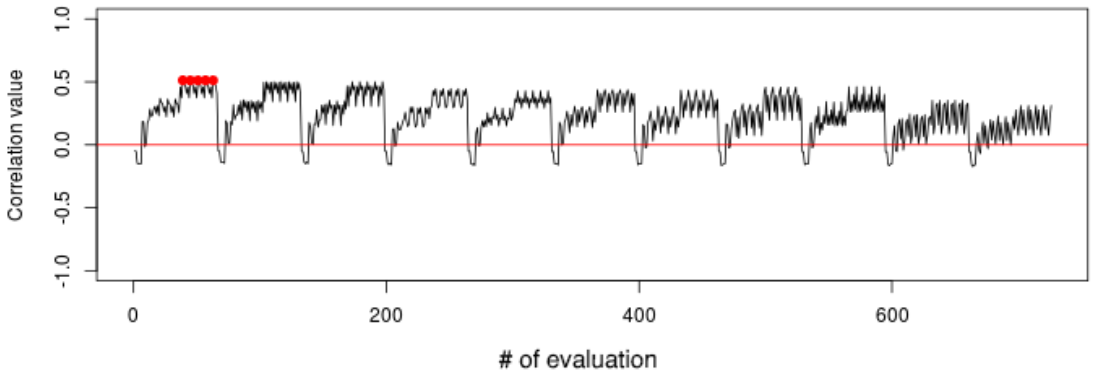


Figure 4.5: Correlations of the modes' people answers and final results' system

holding all the parameters fixed except one. As is shown, the $Threshold_{SRC}$ and *zeroRatio* parameters are variables that create some significant dynamics in the values in opposite ways, i.e, when the former inscreases the correlation value is lower, while when the last increases the correlation grows. The other parameters seem to have a lesser effect in the results but they all make a contribution in the end, when they're all combined.

Four of the five parameters were already determined with all the correlations and the analysis done previously. The last parameter, the *Threshold*, that defines whenever a description is good or not, was estimated through a F-measure based optimization. In table 4.1 are presented all the values with the *Threshold* varying between 0.01 and 0.3, limited by the precision or the recall reaching 1. Based on this results, the best F-measure value was 0.72 for a *Threshold* of 0.14. Its precision-recall shows that the system finds almost all the positive values with a recall of 93% and has a precision of 59%. Both values seemed to be the most balanced, considering the nearest F-measure values, 0.70, where the recall is the same but the precision is lower than the one with the *Threshold* of 0.14.

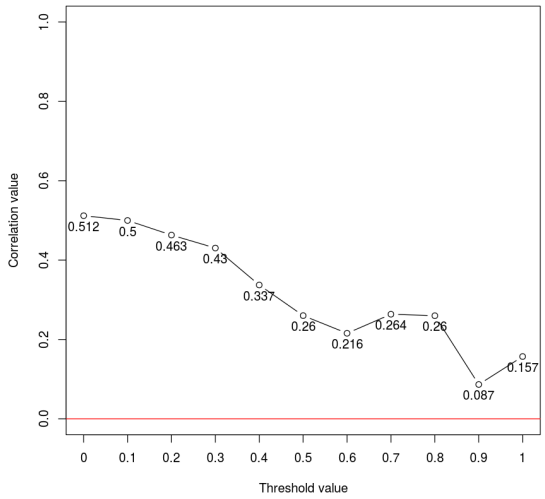
In the same part of the survey, the first question is responded with the hypothesis

People's classification and the system's evaluation are related?
$$H0 : \rho = 0 \text{ vs } H1 : \rho \neq 0$$

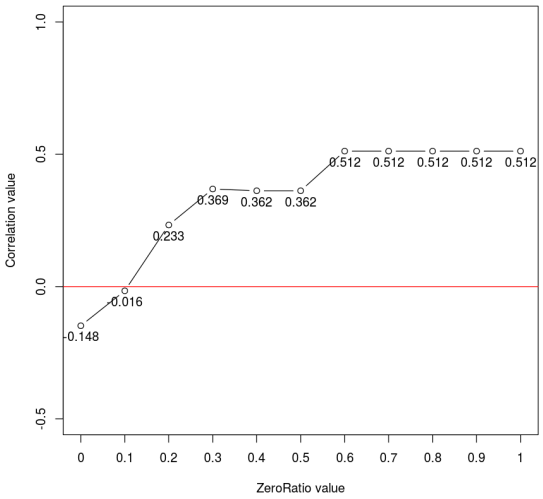
to the Spearman's correlation coefficient test, ρ_s, where

$X =$ *random variable which represents an image value given by the system*,
$Y =$*random variable wich represents the average of the answers given by the people of an image*.

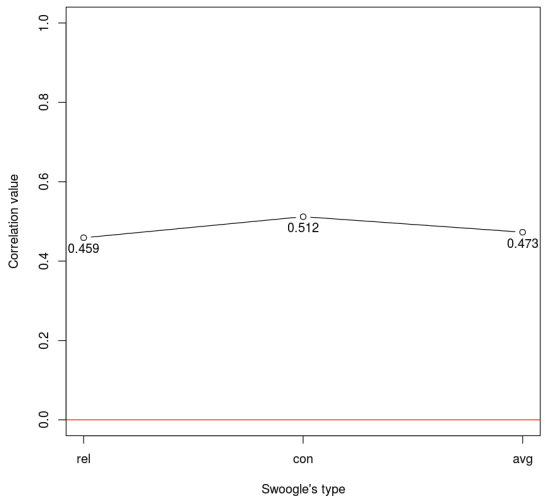
Based on the results, it can be settled that both variables are related, as a result of all runnings being always $\rho \neq 0$ and the best corelation $\rho = 0.5811$ with a $p - value = 0.0007581$. This rejects $H0$, with a moderated correlation [43] , therefore we have some evidence that people's classification and the system's evaluations values are monotically



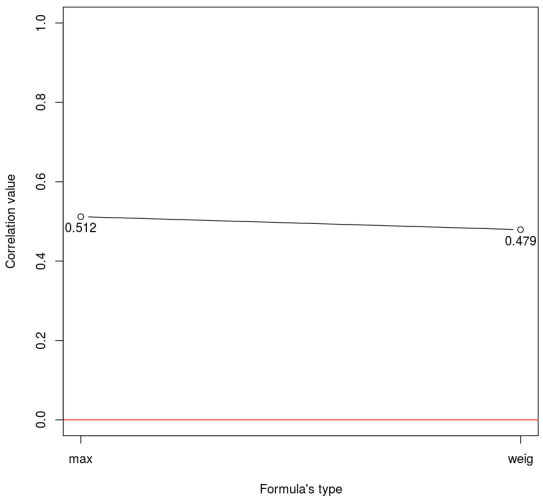
(a) $\text{Threshold}_{\text{SRC}}$



(b) ZeroRatio



(c) Swoogle's type



(d) Formula's type

Figure 4.6: Result for each parameter while holding the others

correlated, which means the results of the system are somehow close to the people's answers.

Additionally, further correlations were made to understand the weight of the domains presence in the evaluation, as can be seen in the table 4.2. Three domains were tested, *persons*, *places*, *organizations* and all combinations between them. The first detects any person noun in the expression, the second any places noun and the third any organization noun. Based on the results there is an interessant outcome for the domains combinations. First, it's conclusive that there are domains stronger than others, for example the correlation is lower when the domain *organizations* is inserted and in contrast, the domain *persons* enhances a better result. This situation can be substantiated with the fact

Threshold	Precision	Recall	F-Measure
0.00	0.48	1.00	0.65
0.01	0.48	1.00	0.65
0.02	0.48	1.00	0.65
0.03	0.48	1.00	0.65
0.04	0.48	1.00	0.65
0.05	0.48	1.00	0.65
0.06	0.50	1.00	0.67
0.07	0.50	1.00	0.67
0.08	0.48	0.93	0.63
0.09	0.50	0.93	0.65
0.10	0.50	0.93	0.65
0.11	0.52	0.93	0.67
0.12	0.57	0.93	0.70
0.13	0.57	0.93	0.70
0.14	**0.59**	**0.93**	**0.72**
0.15	0.57	0.86	0.69
0.16	0.50	0.57	0.53
0.17	0.62	0.57	0.59
0.18	0.58	0.50	0.54
0.19	0.55	0.43	0.48
0.20	0.60	0.43	0.50
0.21	0.62	0.36	0.45
0.22	0.60	0.21	0.32
0.23	0.50	0.14	0.22
0.24	0.50	0.14	0.22
0.25	0.67	0.14	0.24
0.26	0.67	0.14	0.24
0.27	0.67	0.14	0.24
0.28	0.67	0.14	0.24
0.29	1.00	0.07	0.13
0.30	1.00	0.07	0.13

Table 4.1: F-measure calculations

that some domains can be more frequent on descriptions than others. So, answering the second question, there are different results depending on the used domains. Explicitly the best correlation is 0.581, when the *persons* and *places* domains are combined. The worst is *organizations* with 0.276.

The third question is similar to the first one but related with the words used in the second part of the survey. As was said, these words are considered the best descriptors by Clarifai, so the question is about how well these words describe the images being evaluated. For this was made an average of all words' classification which describe the image, used in the form, seen in the figure 4.7. Thus, we propose the following hypothesis

Is the mean of all words' classification above of the sample mean?

$$H0 : \mu_0 \leq 2.49 \text{ VS } H1 : \mu_0 > 2.49$$

Table 4.2: Correlations with domains

Domains	ρ
No Domain	0.371
persons	**0.512**
places	0.346
organizations	0.276
persons + places	**0.581**
persons + organizations	0.441
places + organizations	0.284
persons + places + organizations	0.501

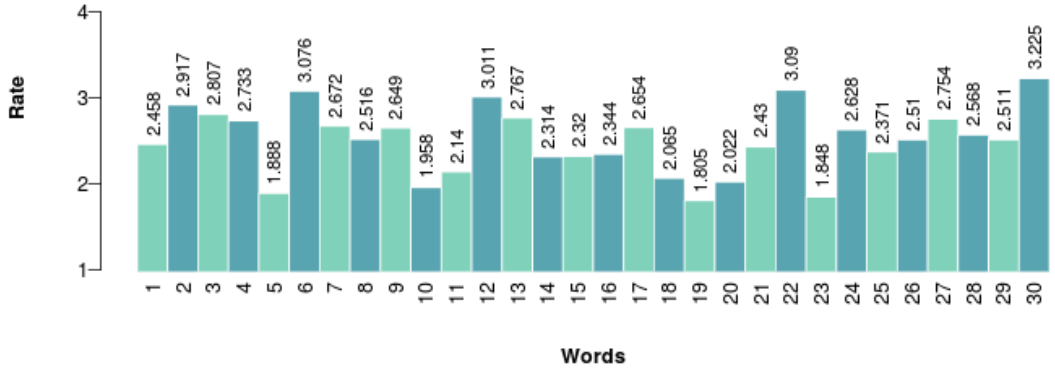


Figure 4.7: Mean of words' classification of participants, between 1 and 4

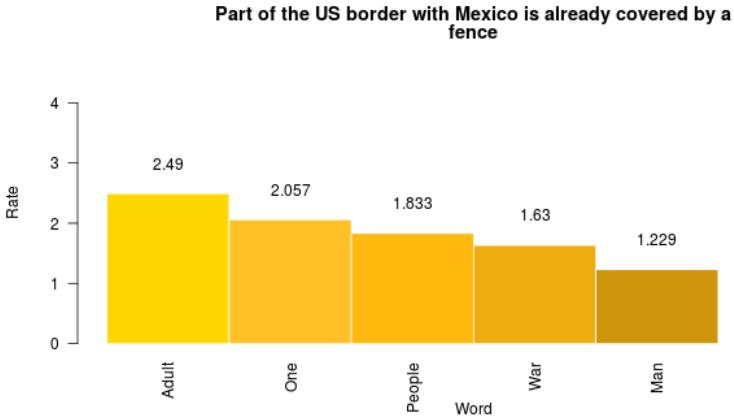
Considering a total of 30 images, with the population standard deviation unknown, σ , the sample standard deviation estimated to $S = 0.378$ and the sample mean $\bar{X} = 2.50$. The statistic test for the mean value goes has followed:

$$Z = \frac{\bar{X}-\mu_0}{S/\sqrt{n}} \frown^a N(0,1) \text{ if } H0 \text{ is true}$$
$$z = \frac{2.50-2.49}{0.383/\sqrt{30}} = 0.24$$

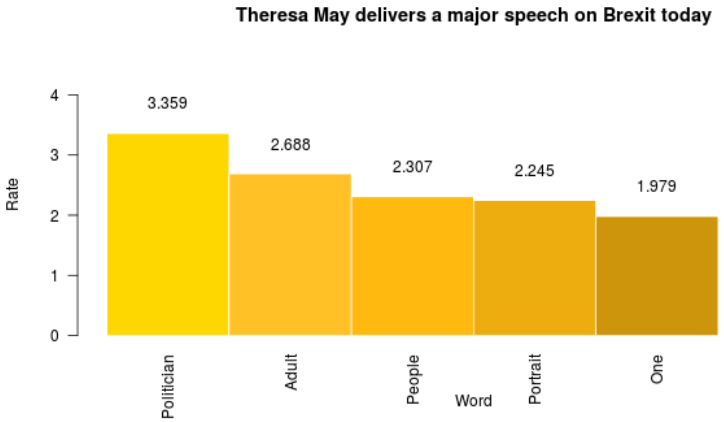
resulting in

$$p - value = P(Z \geq z) = 1 - \phi(z) = 1 - \phi(0.24) = 1- \sim 0.595 \simeq 0.405$$

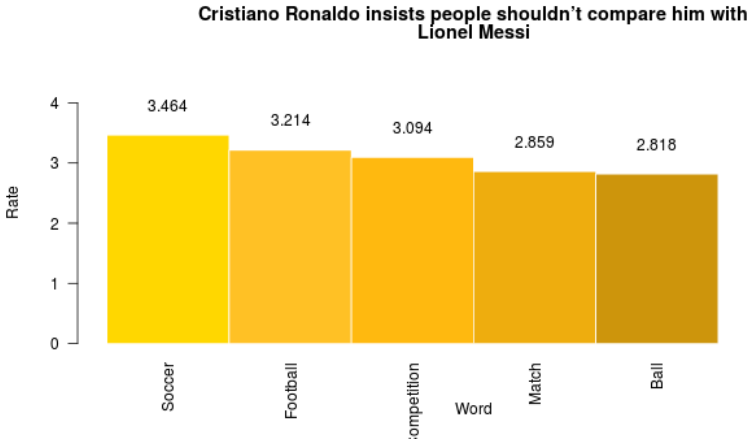
The population doesn't follow a normal distribution, as was previously verified. Afterall, it's not possible to reject $H0$, because there are no evidences to prove that the mean value of the words is superior to the sample mean, with a $p - value = 0.405$. Therefore, it's inconclusive that these words are good descriptors for the images. Following the same figure, all the values seem to be very inscontant and weak. There are opposite situations where the words have really high values (figure 4.8c) and others that the values are lower (figure 4.8a) or very inconstant (figure 4.8b).



(a) Low ratings



(b) Mixed ratings



(c) High ratings

Figure 4.8: Example's of words' ratings

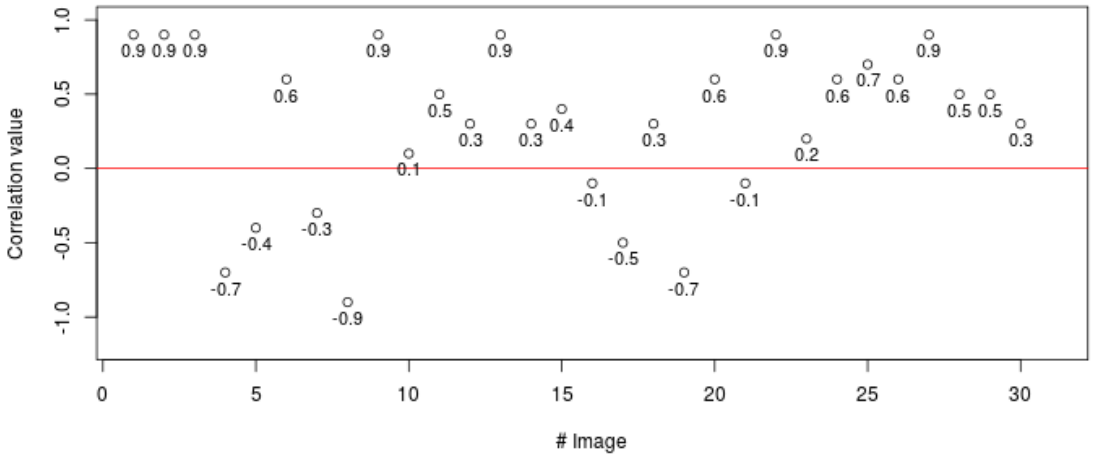


Figure 4.9: Scatter plot of the correlation between original and people indexes

To answer the fourth question and to understand if the order of the words is the same in the Clarifai as in the participant's answers, an evaluation comparing indexes was made. Originally, the words are kept with the Clarifai's order which mean their indexes will represent the basis vector [1,2,3,4,5]. Each set of words, classified by participants, will have other indexes' vector. To know these new positions, a comparison between both words' vectors is executed. After getting the vector indexes of participants, a correlation is measured, as can be seen in the figure 4.9, with a mean of $\rho = 0.303$. This value represents some association between the two vectors. Even if being weak, both orders can be considered somewhat similar.

Example:
ClarifailOrder - [”cat”,”dog”,”fish”,”horse”]
ClarifaiIndexes - [1 ,2 ,3 ,4]
ParticipantsOrder - [”fish”,”cat”,”dog”,”horse”]
ParticipantsIndexes() - [2, 3, 1, 4]
$\rho = 0.4$

In this example, there is an original order for the words (this represents Clarifai's order), then with the ParticipantsOrder, where the only word kept in the same position was ”horse”, the correlation was 0.4. If the words are all in the opposite order, like [4,3,2,1], the value is -1, with the correlation representing an association, but its signal says that they're in the exact inverse order.

Finally for the last question, it's calculated a mean of all responses given by participants. Since the sample mean is 2.5, any value above it is considered a good classification. In table 4.3 are shown the results of the participants classifications of every alternative text. The first column corresponds to the images' number of the table in appendix A. The next four columns are the scale between 1 and 4 with the total of answers in each rating.

The last column is the average of these classifications of each image. The mean of these ratings is 2.5, equals to the scale mean, there are 15 values equal or above 2.5, which means half of this set has good descriptions and the other half bad descriptions. Therefore, its no possible to conclude if all the original alternative texts are considered good descriptions according to the participants' ratings. Nevertheless, we can conclude that there are many original alternative texts which are classified as bad, which means there are images in the Web that are being poorly described, misleading the disabled users.

Table 4.3: Participants classifications

image number	1	2	3	4	avg
1	13	61	79	39	2.75
2	109	61	18	4	1.57
3	6	34	77	75	3.15
4	2	32	82	76	3.21
5	6	20	56	110	3.41
6	158	23	8	3	1.25
7	6	9	41	136	3.60
8	14	70	74	34	2.67
9	49	56	47	40	2.41
10	137	36	16	3	1.40
11	40	62	60	30	2.42
12	51	75	42	24	2.20
13	3	5	16	168	3.82
14	38	50	52	52	2.61
15	95	58	29	10	1.76
16	3	22	60	107	3.41
17	135	42	11	4	1.40
18	102	67	13	10	1.64
19	66	88	25	13	1.92
20	6	14	28	144	3.61
21	7	32	79	74	3.15
22	94	66	24	8	1.72
23	22	50	88	32	2.68
24	117	38	26	11	1.64
25	3	9	32	148	3.69
26	137	39	9	7	1.41
27	87	80	21	4	1.70
28	3	10	55	124	3.56
29	96	59	27	10	1.74
30	3	22	41	126	3.51

4.2 Final Version

In sum, the algorithm operates over three entities that were previously described: the description, the concepts and the domains. Then, three relations can be established between them. One relates the description with the image's concepts (*SCDesc*); other relates the description with each existing domain (*FDD*) and the last one relates each domain and each concept (*SCDom*). In the pseudocode 2 there is the algorithm of Screw after being optimized. Based on the results of the study, the best values were considered as:

1. $Threshold_{SRC}$ - 0

2. *zeroRatio* - 0.6
3. *Swoogle's type* - Concept
4. *Formula* - Maximum
5. *Threshold* - 0.14

Algorithm 2 Algorithm to be adjusted

Input: $concepts, description, domains$

for $domain$ in $domains$ **do**
    GDD [$domain$] ← Calculate general SR ($description$, $domain$)
    SDD [$domain$] ← Calculate specific SR ($description$, $domain$)
    FDD [$domain$] ← Calculate final SR($description$, $domain$,$SDD[domain]$,$GDD[domain]$)
end for
for $concept$ in $concepts$ **do**
    SCDesc[$concept$] ← Calculate SR ($concept$, $description$)
    for $domain$ in $domains$ **do**
        SCDom[$concept$,$domain$] ← Calculate SR ($concept$, $domain$)
        if Exist some value in FDD[$domain$] **then**
            SCDD[$concept, domain$] ← Calculate **MAX** (SCDom[$concept$,$domain$], SCDesc[$concept$], FDD[$concept$,$domain$])
        end if
    end for
end for
$avgAllconcepts$ ← Calculate average of all SCDD values > 0
$zRatio$ ← Calculate ratio of zeros in SCDD
if $zRatio >=$ **0.6 then**
    $finalValue \leftarrow 0$
else
    $finalValue \leftarrow avgAllconcepts$
end if
if $finalValue <$ **0.14 then**
    $verdict$ ← false
else
    $verdict$ ← true
end if

Output: $finalValue$ and $verdict$

For each concept and each domain there is a value which represents their semantic relation. If the description and the concept are related with the same domain, i.e. both relation values are above zero, their semantic relation will be strengthened, then a new relation value between them is calculated. This new value will be the maximum between the following relations, concept with description(*SCDesc*) and concept with domain (*SCDom*). After calculating all the new relation values, the algorithm calculates an average

Figure 4.10: Algorithm's example

of all positive values (≥ 0), i.e those that are meaningful, if the semantic relation between the description and the domain (*FDD*) has some value. This average represents the semantic relation value between the content and the description. Besides this, all concepts that don't have any relation with the description, i.e. their value is zero, will be considered for the final result as well. If the amount of zero values is bigger than a certain percentile (0.6), then the relation between the content and the description is considered a bad one. The algorithm's output (*finalValue* and *verdict*) will be the *avgAllConcepts* value, which represents the value of the semantic relation between the description and the content, and the *verdict* value. The latter is defined according to a threshold (0.14), which dictates if the description is good (true) or bad (false), depending on the *avgAllConcepts* value being above or below that threshold.

4.2.1 Example

Using the eighth image of appendix A as an example, all entities and variables will be showed with real values. Note that the evaluation was reduced to only 10 concepts, just for the example.

Figure 4.10 has "*Theresa May delivers a major speech on Brexit today*" as its description (alternative text). Table 4.4 shows the results of the relation between the description and the two existing domains, *persons* and *places*. Table 4.5 shows all the relation values between the concepts and, the domains and the description; the first column has all the image's concepts, the second column has all the raw values from the relation between the description and the concepts, the next two columns are the values between each domain and each concept and the last one is the final results after the maximum formula been applied.

In this case, the description has a non-null relation with each domain, which allows to execute the maximum between the *SCDesc* and *SCDom* for each concept. The *finalValue* is 0.2644003, since the amount of zero values in *SCDesc* is only one which gives a ratio of 0.1, lower than 0.6. The final result is 0.2644003 with a verdict of *true*, because 0.2644003 >= 0.14.

4.3 Summary

A study was carried and reported in this chapter to determine the values of the algorithm's parameters of the evaluation module, so that the algorithm could perform in the

Table 4.4: Description and domains - semantic relation

	DOMAINS	
	persons	places
description	0.68660731	0.03381388
	FDD	

Table 4.5: Semantic relations with concepts, domains and description

	SCDESC	**DOMAINS**			**SCDD**
CONCEPTS	description	persons	places		description
portrait	0.047559194	0.048137933	0.045044		0.048137933
one	0	0	0		0
people	0.042820755	0.5988054	0.16138849		0.5988054
adult	0.015457052	0.6743183	0.03808342	**MAX ()**	0.6743183
politician	0.021817537	0	0.017570926		0.021817537
woman	0.023719406	0.6842941	0.0027263092		0.6842941
face	0.034089174	0.09488924	0.053856425		0.09488924
business	0.029714782	0.082308665	0.054849		0.082308665
festival	0.013783761	0	0.10153505		0.10153505
leader	0.05021154	0.07349613	0.070952505		0.07349613
		SCDOM			

best way. Then, the final version of the algorithm, after optimization, is described with an example complementing the definition to ease its understanding.

Chapter 5

Integration

Screw was developed to achieve two of the objectives of this work which are improving accessibility with Qualweb and aid users to browse the Web with an automatic repair mechanism. However, with Screw developed as a stand-alone system it's not so helpful, since it only provides the result of an evaluation or a suggestion from the repair mechanism. It's convenient to use these results to resolve some issues mentioned in chapter 2. For the first objective was necessary to understand the exact point of Qualweb's assessments regarding semantic content. With a more conscious perspective about Qualweb's state, we incorporate Screw as a complement to supress some lack of quality evaluation as will be reported in this chapter. For the second objective, a plugin was developed for the Chrome browser in order to repair automatically the DOM of the Web pages. In figure 5.1 there is a schema illustrating how the components are connected. Qualweb, as explained before, can perform evaluations through the command-line or within the browser. The plugin only works with Chrome browser, which performs HTTP requests to Qualweb, which in turn evaluates the DOM, including already the semantic assessment. The whole process will be detailed in the following sections.

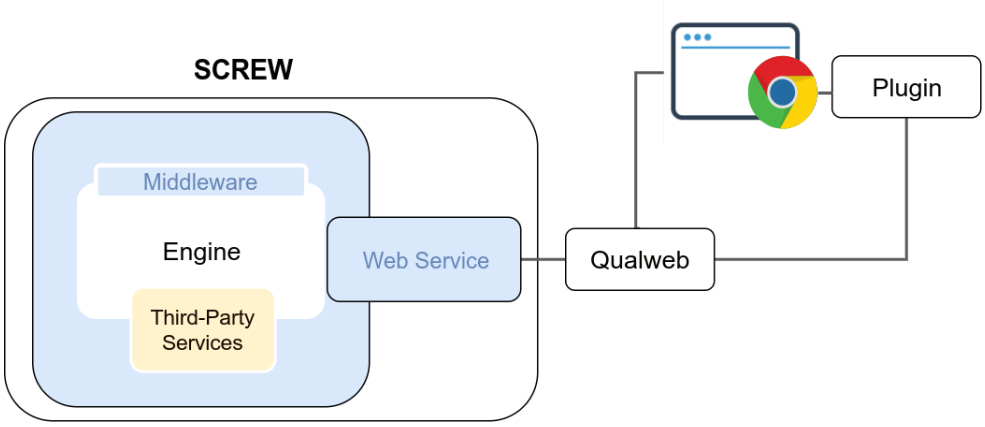


Figure 5.1: Integrations architecture

Table 5.1: Qualweb techniques with semantics

Techs	Description
h2	Combining adjacent image and text links for the same resource
h25	Providing a title using the title element
h30	Providing link text that describes the purpose of a link for anchor elements
h33	Supplementing link text with the title attribute
h36	Using alt attributes on images used as submit buttons
h37	Using alt attributes on img elements
h39	Using caption elements to associate data table captions with data tables
h44	Using label elements to associate text labels with form controls
h45	Using longdesc
h53	Using the body of the object element
h54	Using the dfn element to identify the defining instance of a word
h57	Using language attributes on the html element
h64	Using the title attribute of the frame and iframeelements
h65	Using the title attribute to identify form controls when the label element cannot be used
h73	Using the summary attribute of the table element to give an overview of data tables
h89	Using the title attribute to provide context-sensitive help
h90	Indicating required form controls using label or legend

5.1 Qualweb

5.1.1 Semantic Content with WCAG2.0

In the Related Work chapter, Qualweb was fully characterized, from its architecture to its execution, including a list of all its developed techniques (from WCAG2.0). During the analysis no technique was found that could perform a semantic analysis. It was concluded that Qualweb doesn't have this kind of evaluation. To suppress this flaw, we integrated Screw in Qualweb. For Screw to be useful it's important to know which techniques demand a semantic assessment.

Table 5.1 shows a set of techniques implemented in Qualweb that rely on semantics. These techniques don't just treat semantic aspects, their procedures also include syntactic points, which is what was performed by Qualweb until this stage.

In most cases, these techniques require to check the description or the content of images, anchors, tables, links, form controls and others. They have attributes that can represent a description, like alternative text (alt) or a title, or through other type of tags (legend, fieldset, captions and others). Techniques like h42, h43, h63, h69, h70, h75, h85 and h97, which are related to the elements' inter relations in the Web page's structure, will be not considered in this work.

The techniques which are shown in table 2.2 are divided in two types of assessment:

text and image. Since the main focus was evaluating images, the h37[1] technique was considered for the integration of Screw in Qualweb. Despite being just one technique, the issue which is handled is quite frequent and rarely corrected.

H37 Procedure

1. Examine each img element in the content
2. Check that each img element which conveys meaning contains an alt attribute.
3. If the image contains words that are important to understanding the content, the words are included in the text alternative.

Expected Results

- Checks #2 and #3 are true.

5.1.2 Integrating H37

Since the beginning, Screw was planned to work closely with Qualweb. However, during the development process some incompatibilities were found, which wouldn't allow communication between both tools. Qualweb was developed over a Node.js engine with its funciontingbased on asynchronous calls, but programmed in a different way (without using callbacks) from how Screw has been developed (using callbacks). This created some technical issues that wouldn't allow merging both tools. With the development of the webservice this issue was mitigated.

As shown in the previous table, the technique h37 is already partially implemented in Qualweb, with only syntax evaluation. Therefore, it was only need to modify this technique file to call Screw's webservice to check the description. For each *img* tag it's source (*src*=) and alternative text (*alt*=) attributes are retrieved to get the inputs to be evaluated by Screw, the image's URL and description.

Before calling Screw, the image's URL needs to be verified, because often the source attribute comes with an incomplete URL, without the hostname, and when it comes with the hostname sometimes the "http" is missing. Also, they might come with different formats like starting with "data:" or ending as ".gif". When these last two restrictions happen, the image is not considered for evaluation.

After Screw's evaluation, its results will define if the technique is successfull or not, according with the Screw's verdict, true or false. This technique outputs three different states: pass, warning or failed. The first happens when Screw gives a verdict of true, i.e., the alternative text is a good description for the image. The second one happens when the alternative text exists, but Screw gives a verdict false value and the technique fails, when there is no alternative text or when it comes with a single word like "image" or

[1] https://www.w3.org/TR/WCAG-TECHS/H37.html

"picture". For the last two states, the technique sends both needed inputs for the Screw's repair mechanism.

After being evaluated, if the result of the element is failed or warning, the element will be repaired with Screw, giving a new description, further in the repair module of Qualweb. The final output of Qualweb is all the evaluation results, including the repair items and results, that will be used by the frontend page of Qualweb or by the plugin.

5.2 Plugin

The objective of this plugin is to repair automatically the Web pages in order to amplify their accessibility through evaluations from Qualweb and Screw. This plugin is developed only for Chrome browser, because previously there was a plugin already implemented for this environment and to leverage its implementation the new plugin remains in the same context.

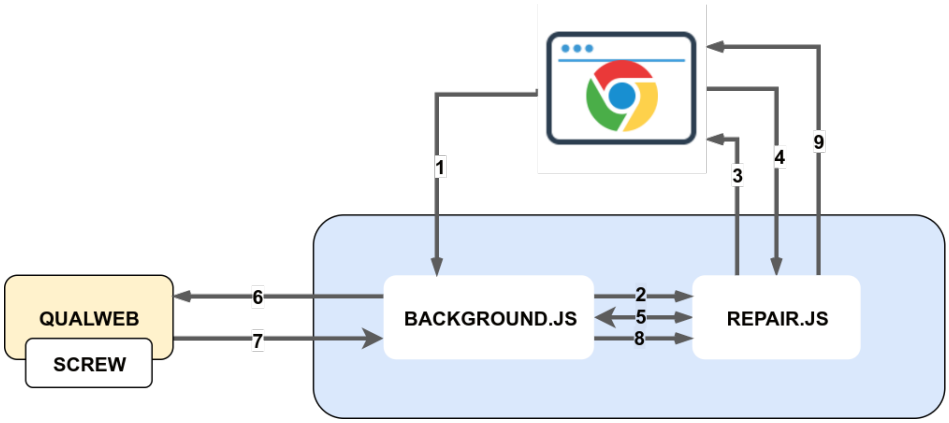


Figure 5.2: Plugin's schema

The plug-in is composed by two files, background.js and repair.js. In figure 5.2 these components are shown together with the communication between them and the tools, the Chrome browser and Qualweb/Screw. The former is always listening the pages, i.e. checking if there is any new tab with a valid URL and what tab is the "current" one and if the current page is refreshed or deleted. The background file also deals with the communication between the plugin and Qualweb. It stores in the LocalStorage some useful variables to identify which URL is in the open tab and if it's being evaluated or not. When the Webpage is refreshed or it's just a new one (**1**), the background will send a message to the repair.js, saying that the page is ready to handle (**2**). Then the repair script will send the new DOM to the background (**5**) which in turn will request an evaluation to Qualweb (**6 and 7**). The result is sent back to the repair script (**8**), which verifies for each technique the elements with failed or warning results and repairs them (**9**). There are three main states in the plugin: not evaluating, evaluating and evaluated. Not evaluating means that the current page isn't readable, doesn't have a valid URL to be evaluated. The second

(a) Not evaluating (b) Evaluating (c) Evaluated

Figure 5.3: Plugin's states

means that the current page is being evaluated and the last one represents a concluded evaluation. The first and last states are followed by an audio warning.

The repair.js is a content script, a specific script established by Google, which has permissions to read the page's DOM. This is set up inside the *manifest.json*, herewith other configurations to grant permissions to get data about the tabs. Originally an evaluation is made usgin the URL. However differences were found between DOM's from different sources. So, instead of asking for an evaluation in the background by the URL, the DOM actually captured by the content script is sent, because this will represent the closest DOM that is being interpreted in that instant by the browser (**3 and 4**). This avoids evaluating different DOM's which would cause problems, when comparing the results with the actual DOM because the position of the elements will be different. The ID's of the elements must be the same on both sides, when evaluated by Qualweb and when the original is going to be repaired. The sent DOM is previously altered by inserting IDs to ensure that each element is unique. This facilitate a further comparison between the original DOM and the elements evaluated. Once again, they must be equal.

5.2.1 Example

Listing 5.1 is an example of a page with only four images, two of them have an alternative text with something within, the third one has an alternative text but it's empty and the last one doesn't have it. Figure 5.4 shows the page (with the pictures only), the first one is a field with a person collecting or checking some plants, the second one is a celebrity, who's name is Beyonce, carrying two babies, the third one are two lions and the last one is the same as the first (but with no alternative text). This example shows a diversity of situations, when the images have an *alt* atribute, when it's filled or empty, and when it doesn't have one.

Figure 5.4: Plugin images of example

Figure 5.5 has two images where the code is presented. These snippets were took inside the browser inspector. The first one shows the original code with the new IDs inserted by the plugin, which you can see it's different from the original code (no ele-

ment has IDs). The last one is the result of the repair, which took around 47 seconds to evaluate. Table 5.2 describes the original states and repaired sentences of the alternative texts of each image. Only the three last were repaired, the first one was considered an adequate text so it remains the same. The second one was emended with a sentence from CaptionBot, because having only a name of a person is not considered a good alternative text since it depends of culture matters. In these type of situations the sentences should be more completed, perhaps merging both alternative texts should help, like “Beyonce, a woman standing in front of a flower...". The last two pictures were corrected because one had a empty alternative text and the other was lacking it. Although an empty alternative text means that sometimes images can be decorative, knowing it was not considered.

```
1  <html>
2 <head>
3    <title> h37 test </title>
4 </head>
5 <body>
6
7    <img width=''23%'' src=''http://www.telegraph.co.uk/content/
         dam/investing/2017/07/14/TELEMMGLPICT000108708117-
         large_trans_NvBQzQNjv4BqpVlberWd9EgFPZtcLiMQfyf2A9a6I9Ychs
         jMeADBa08.jpeg'' alt=''Tobacco fields''>
8
9    <img width=''23%'' src=''http://www.telegraph.co.uk/content/
         dam/technology/2017/07/14/beyonce-new-instagram-
         large_trans_NvBQzQNjv4Bq9WvEVMUXXRGVdwl10TLNlhRY9bnFVTp4QZlQ
          jJfe6H0.jpg'' alt=''Beyonce''>
10
11   <img width=''23%'' src=''http://www.awf.org/sites/default/
         files/media/gallery/wildlife/Lion/ Federico_Veronesi_2009-01
         -28%20Masai%20Mara_4728.jpg?itok=96CysP6Y'' alt=''''>
12
13   <img width=''23%'' src=''http://www.telegraph.co.uk/content/
         dam/investing/2017/07/14/TELEMMGLPICT000108708117-
         large_trans_NvBQzQNjv4BqpVlberWd9EgFPZtcLiMQfyf2A9a6I9Ychs
         jMeADBa08.jpeg''>
14
15 </body>
16 </html>
```

Listing 5.1: Plugin example’s code

5.3 Limitations

In production time was found that each page could take several minutes, up to 20 minutes, to be evaluated. This is a problem that must be resolved in order to have a faster system and to not annoy users with a long waiting time. Another issue is the lack of repair techniques. Qualweb only has the h37 technique as well as the plugin, which is very incomplete compared with the amount of techniques that are already implemented

```
<!DOCTYPE html> == $0
<html>
▶#shadow-root (open)
▶<head>…</head>
▼<body id="ubiwan_generated_id_1">
    <img width="23%" src="http://www.telegraph.co.uk/content/dam/
    investing/2017/07/14/TELEM…7-
    large_trans_NvBQzQNjv4BqpVlberWd9EgFPZtcLiMQfyf2A9a6I9YchsjMeADBa08
    .jpeg" alt="Tobacco fields" id="ubiwan_generated_id_2">
    <img width="23%" src="http://www.telegraph.co.uk/content/dam/
    technology/2017/07/14/beyo…am-
    large_trans_NvBQzQNjv4Bq9WvEVMUXXRGVdwl10TLNlhRY9bnFVTp4QZlQjJfe6H0
    .jpg" alt="Beyonce" id="ubiwan_generated_id_3">
    <img width="23%" src="http://www.awf.org/sites/default/files/media/
    gallery/wildlife/Lion/Federico_Veronesi_2009-01-
    28%20Masai%20Mara_4728.jpg?itok=96CysP6Y" alt id=
    "ubiwan_generated_id_4">
    <img width="23%" src="http://www.telegraph.co.uk/content/dam/
    investing/2017/07/14/TELEM…7-
    large_trans_NvBQzQNjv4BqpVlberWd9EgFPZtcLiMQfyf2A9a6I9YchsjMeADBa08
    .jpeg" id="ubiwan_generated_id_5">
  </body>
</html>
```

(a) Original DOM

```
<!DOCTYPE html> == $0
<html>
▶#shadow-root (open)
▶<head>…</head>
▼<body id="ubiwan_generated_id_1">
    <img width="23%" src="http://www.telegraph.co.uk/content/dam/
    investing/2017/07/14/TELEM…7-
    large_trans_NvBQzQNjv4BqpVlberWd9EgFPZtcLiMQfyf2A9a6I9YchsjMeADBa08
    .jpeg" alt="Tobacco fields" id="ubiwan_generated_id_2">
    <img width="23%" src="http://www.telegraph.co.uk/content/dam/
    technology/2017/07/14/beyo…am-
    large_trans_NvBQzQNjv4Bq9WvEVMUXXRGVdwl10TLNlhRY9bnFVTp4QZlQjJfe6H0
    .jpg" alt="A woman standing in front of a flower,and she seems to
    have a neutral face.
    " id="ubiwan_generated_id_3">
    <img width="23%" src="http://www.awf.org/sites/default/files/media/
    gallery/wildlife/Lion/Federico_Veronesi_2009-01-
    28%20Masai%20Mara_4728.jpg?itok=96CysP6Y" alt="This image is mainly
    about these concepts: lion, cat and mammal
    " id="ubiwan_generated_id_4">
    <img width="23%" src="http://www.telegraph.co.uk/content/dam/
    investing/2017/07/14/TELEM…7-
    large_trans_NvBQzQNjv4BqpVlberWd9EgFPZtcLiMQfyf2A9a6I9YchsjMeADBa08
    .jpeg" id="ubiwan_generated_id_5" alt="Person in a garden.
    ">
  </body>
</html>
```

(b) Repaired DOM

Figure 5.5: Example of a repair through plugin

Table 5.2: Plugin result

Image	Original Alt	Repaired	Repaired Alt
1	Tobbaco fields	FALSE	–
2	Beyonce	TRUE	A woman standing in front of a flower,and she seems to have a neutral face.
3	<empty>	TRUE	This image is mainly about these concepts: lion, cat and mammal
4	NA	TRUE	Person in a garden

in Qualweb.

Other issue about Qualweb and the plugin is concerned to the requests made in each browser's tab. It's difficult to understand if the oldest evaluations being performed are being stopped or inserted in a buffer (waiting for others finishing) when a new page is loaded, because there is no evaluation state coming from Qualweb.

5.4 Summary

This chapter presented two integrations of Screw.

The first tool is Qualweb, a Web page evaluator for Web accessibility, which verifies if the pages are according to the rules presented in WCAG2.0 guidelines. Screw's integration was supported through WCAG2.0 techniques that demand semantic evaluations of images.

The second integration was established with a plugin whose purpose is to evaluate and repair pages in real time, allowing impaired people to browse the Web with lower restritions. For each page, the plugin makes requests to Qualweb which in turn requests assessments to Screw. According to the results, the plugin repairs the page through DOM manipulation.

Chapter 6

Evaluation

This chapter presents an evaluation of the system's results to understand their quality. The setup of this survey is similar to the one done in section 4.1, using a set of images and their alternative texts. Not only the evaluation module will be analysed but also the quality of the descriptions generated by the repair module. We used an online survey, as well. The following research questions were considered:

1. Is the algorithm, with optimized parameters, still reliable with a different set of images?

2. Does the CaptionBot give good repair sentences?

3. Is the "3 words sentence" of Clarifai a good repair sentences?

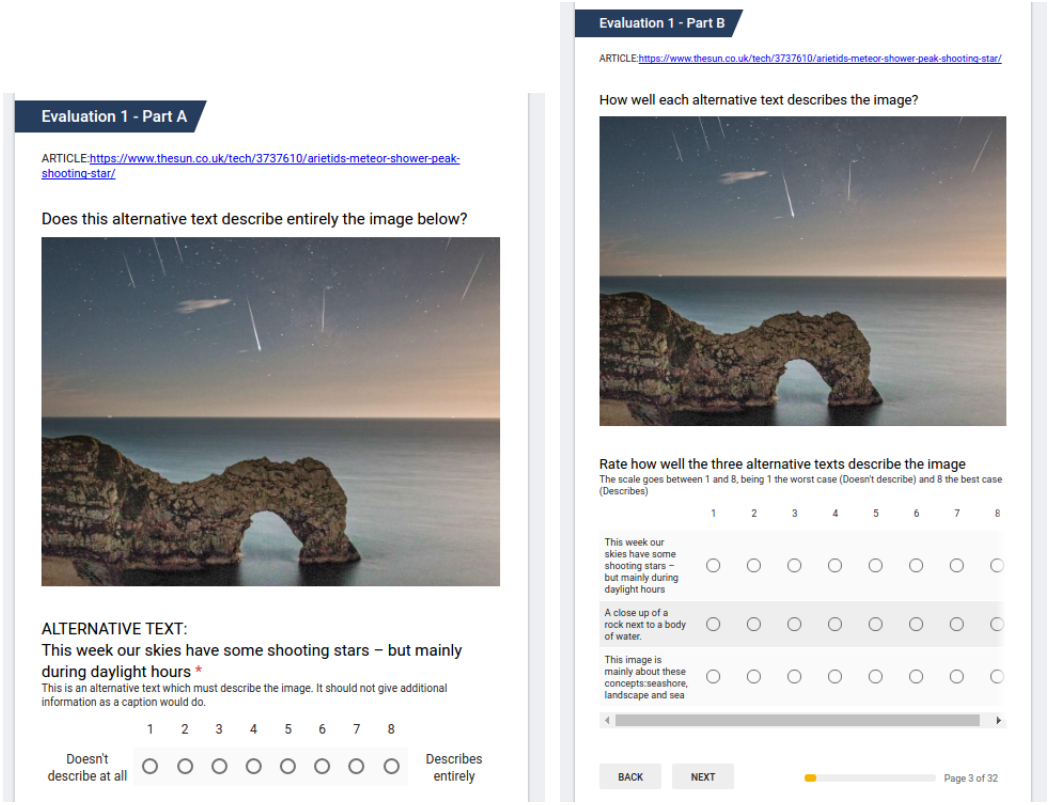
6.1 Setup

For this survey, fifteen images with their own alternative texts were collected. This images are all different from the first study to validate the system after the adjustments made. The requirements are the same of the first, with all of the alternative texts in English and also from real contexts. They were retrieved in the same sources: The Telegraph and The Sun. A diverse criteria was followed collecting images of persons, landscapes and objects. In appendix C there is a table with all the images with the respective alternative text.

This form was also shared via e-mail (mailing lists of institutions and universities) and social networks. We expect most of the survey respondents to not have English language as native and therefore we asked their proficiency level. Since answers are anonymous, there is no way to connect them to the first survey held before.

The survey, as in the first, was developed with Google's Form and is divided in two parts as well (figure 6.1). Part A follows the same structure of the first survey. To find out if the results of people are related or not with the system's results, people had to define if the descriptions are good or bad for the images. The second part, will assess if the repair system provides good replacements for the original alternative texts and will allow

to perceive if people's opinion changes when other descriptions appear side by side. In the first part, participants are asked to rate in a scale, between 1 and 8, how well the original alternative text describes the image. The second part, doesn't only show the original sentence but also two alternatives, one from CaptionBot and the other one that is composed by the best three words gathered by SIRP, both given by the repair module. In the system, the repair only reproduces one sentence as an output, but for this study, all are shown to evaluate each one individually. The original description is kept in both parts, also to discern if people change their opinion when in the presence of other descriptions.



(a) Second survey's, part A (b) Second survey's, part B

Figure 6.1: Second survey sample

6.2 Results

For this survey, we gathered responses from 101 individuals. This data was analysed with a R script, to compare the values given by the system and the ones inferred from participant's answers. First, the scale 1 to 8 was transformed to one between 1 and 4, to ease further comparisons with the first survey. The reason for using a scale with 8 ratings is to offer a wide range of choices. Since participants are faced with more options in the same context that can influence their classification, this allows to have a more relative position of each sentence's rating.

Most of the people who answered the form had an Advanced proficiency level in

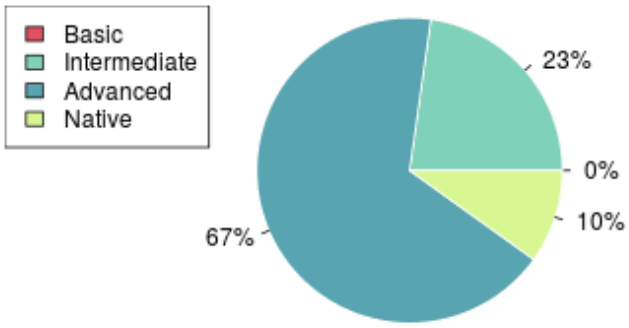


Figure 6.2: Second survey's proffiency level results

English, around 67%. 23% answered as being Intermediate and zero as Basic, unlike the first survey (figure 6.2).

To answer the first research question, new correlations were made with this new set of images. The hypothesis for the population correlation coefficient will be similar to the first survey, which is

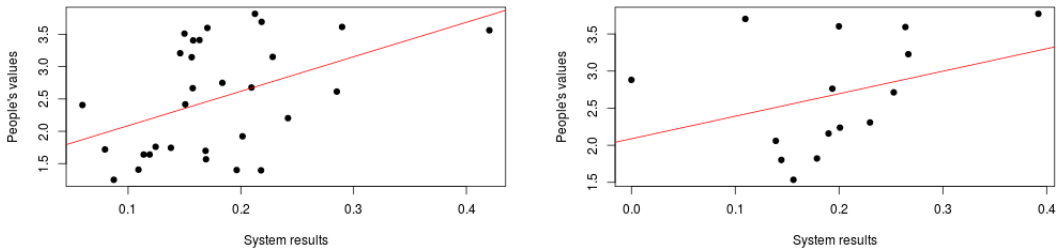
Are the participants answers well related with the system's results?
$$H0 : \rho = 0 \text{ vs } H1 : \rho \neq 0$$

where

$X =$ *random variable which represents an image value given by the system,*
$Y =$*random variable wich represents the average of the answers given by the people of an image*

using Spearman's correlation method. This values can be seen in table 6.1, where in the first and third columns are the system's results of original and CaptionBot descriptions respectively and the remaining are relative to participants' choices of the same sentences. To ease the comparison, the system's values were scaled between 1 and 4, same scale has participants. Additionally, the last resource sentence was also rated by the survey respondents, although the system doesn't make any assessments of those because the concepts were already evaluated and are the three best related with the image.

Based on the results, both variables (*sys:original* and *ppl:original*) are related with a correlation of $\rho = 0.404$, which is less than the resulting value in the former study, but a very positive one. In figure 6.3 two scatter plots are displayed to show the differences between the correlation from survey 1 and survey 2. An interesting circunstance can be seen in the second plot 6.3b, from the last survey, the two first evaluations have opposite results, where the system gave low values and participants gave higher values. These ones are the 8th and the 13 th images in the table C. The former shows an image of a open space with water and a building with its description relying on the name of the hotel and the latest shows a motorcycle with its description relying on its name model. This can be justified with the lack of domains in the system, i.e. the system can not detect names



(a) First survey scatter plot with the adjustments (b) Second survey scatter plot with the adjustments

Figure 6.3: Scatter plots from both surveys

Table 6.1: Second survey's results with system values scaled between 1 and 4

	sys:original	ppl:original	sys:caption	ppl:caption	ppl:resource
1	3.28	**2.31**	4.00	2.54	2.15
2	3.81	3.23	1.52	1.18	2.30
3	2.55	**1.82**	3.90	2.84	2.39
4	**1.99**	**2.06**	**3.69**	3.06	2.29
5	3.77	3.59	2.83	3.40	1.86
6	2.71	**2.16**	3.68	1.83	2.61
7	2.87	**2.24**	2.69	2.25	2.51
8	**0.00**	2.88	**3.37**	1.15	1.95
9	**2.06**	**1.80**	**2.11**	1.10	1.95
10	2.77	2.76	2.79	3.01	1.93
11	2.86	3.60	3.83	1.16	1.57
12	4.00	3.77	4.00	1.53	2.05
13	**1.57**	3.70	**3.73**	1.49	2.18
14	**2.23**	**1.53**	**3.56**	2.62	2.42
15	3.61	2.71	3.88	3.75	2.83
Mean	2.50	2.68	3.31	2.19	1.93

of hotels or of motorcycles brands. This enforces the necessity of having more domains being considered in the evaluation.

For the second and third questions, the table 6.1 shows that people's values are really different from the ones of the system, with low rates in general. For this case, the correlation was also lower with a value of $\rho = 0.232$, quite more negative than the first evaluation accomplished. People think sentences given by CaptionBot are bad descriptors, with a mean value of 2.19, in opposite with system results, with high values and with a mean value of 3.31, as well like the resource sentences, despite having more distributed ratings. The resource sentence had a mean of 2.19, when the sample mean is 2.5, this shows low ratings for this alternative option. Note the threshold that defines the verdict of the evaluation is currently 0.14 and the system gave mostly values above that.

As previously said, the original alternative text was integrated in both parts of the survey. People gave the same rate in both parts, with a correlation of $\rho = 0.998$. This means people don't change their opinion if they see other descriptions that could be better or not than the original.

For the repair module, the system chose to repair the 4^{th}, 8^{th}, 9^{th}, 13^{th} and 14^{th} pictures

(in bold on *sys:original* variable), while people chose 1st, 3rd, 4th, 6th, 7th, 9th and 14th (in bold on *ppl:original* variable). This means that, of the 7 chosen by people, only 3 were considered by the system and there are 2 choices (8th and 13th) that people did not perceive. In column *sys:caption* the sentences values which were chosen by the system are in bold. Comparing these values with people's ratings, there are only 2 sentences (4th and 14th) in 5 that are considered better than the original ones.

6.3 Discussion

The study presented in this section shows that the system can make some good evaluations, with a correlation of 0.4, which is a moderated association with people's ratings. This means that the system is capable of evaluating accessibility issues in Web pages related to image's semantics. However, this value can be improved with new implementations, like inserting new domains in the system, and improving the algorithm with a new study with a bigger set of images.

In the second thread, there is a big difference of the results between the evaluation and repair module. The former had some positive values, not only on the first study,0.51 and 0.58, but also on the second, 0.40. The latest had more negative results, where the sentences of Captionbot had a correlation of 0.23 and a mean of 2.19, and where the resource sentence had a mean of 1.92, lower than the sample mean, 2.5. Additionally, not all of the sentences chosen by the system are equal to people's choices, only 3 out of 7. Also, some choices of the system are not considered by people for repair. One of the big reasons for this discrepancy is due to the invested work in both modules being different. Great part of the work was devoted to the evaluation itself as the primary step to get other mechanisms to work. The repair mechanism only has one service to create sentences through images, CaptionBot, having a poor quantity of options for better sentences. Even with the lack of research for the repair mechanism, there is a solid structure to improve in further studies.

6.4 Summary

In this chapter a study to understand how well the evaluation, after the adjustments, and the repair modules perform was reported . To accomplish this analysis, a survey was run, collecting 101 answers. After analysing the collected data, we were able to answer some research questions about the components and the system functioning, based on participants' ratings. The results showed that the evaluation module is capable of doing accessibility assessments of Web pages and the sentences given by the repair module are, in general, considered bad descriptions by the participants' answers.

Chapter 7

Conclusion

The Web is a convenient tool widely used by people, but there is a group of users with disabilities which has difficulties that common users don't have, while browsing it. Unfortunately, most pages restrict their access with some accessibility issues. This work was developed to resolve some of those problems, specifically about Web content semantics. It's important to ensure that this content is well described, because it gives context to these users. Usually, blind people use assistive tools like screen readers to understand the page. If an image has no alternative text (description) or has a defective one, or even, repeats the caption in the alternative's text place, for example, the user will be misled. To solve this situation some objectives were defined: create a tool capable of assessing semantically the Web content, improve Qualweb's evaluation with this new type of assessment and repair the pages.

Screw is presented as a tool that can perform evaluations and repair web content to improve web accessibility. Its structure is built upon a set of distinct modules distributed by the two main processes, evaluation and repair. Moreover, the system relies in a webservice to collect information about the content under evaluation, which offers interoperability to be used not only by Qualweb but also by other tools. Besides this feature, modularity was another relevant aspect to have in the system, avoiding dependecies between modules and easing further developments of the system.

The algorithm presented is distributed by the evaluation's modules, which are three: Semantic Information Retrieval Processor, Domain Manager and Relations Inspector. The SIRP is responsible to gather summary information about the content, i.e., it gives a set of concepts that are related to the content itself as an interpretation. The DM is responsible to get the semantic similarity between the domains and the description and the concepts given by SIPR. The domains are an important component of the system, because they improve the classification of the semantic relation between the parameters in evaluation. The RI gives the semantic similarity between the description and the concepts, relating these values with the values from the relation between concepts and domains obtained in the DM. The last step of the evaluation is to offer the final result combining the outputs of the prior modules. The descriptor of the content will be positively or negatively classified according to the obtained value being higher or lower than a given threshold. In the

repair part there is only 1 main module which runs several modules to collect a set of sentences according to the given image. Then, each sentence is evaluated and each result is compared to all of them, until the best value has been found. The one with best value will be the new description after repair. Most of this semantic evaluation relies on third-party services, like Clarifai, Indico and Swoogle.

Two surveys were conducted in this work. The first was used to define parameters to optimize the algorithm's performance. The second served to understand the quality of the assessments made by the system. These studies relied on human evaluations of images and their alternative texts, for comparison between their ratings and the system's results. The results show that Screw is well capable to do semantic evaluations, although its repair mechanism showed to be not so efficient. On the other hand, these values were taken by evaluations with existing alternative texts, but in many situations these descriptors do not exist, which even if the repair text is not great, it can still repair web pages when they're missing necessary attributes, so that screen readers can read them with less flaws.

Screw was integrated in Qualweb, which makes it the first web accessibility evaluator to have semantic evaluations. For the repair system, a plugin was developed to transcode the web page source in order to repair it in accordance with Qualweb's and Screw's results. This add-on should provide a real time evaluation to assist users while browsing, but the time to evaluate and repair the Web pages can be very long and so not very practical.

Simultaneously to this work, I conducted related tasks not directly associated with this thesis. One was to run Qualweb evaluations in millions of URLs, through a set of machines set up with Docker. In other task, I developed a part of the new website of Qualweb. Those tasks took me around 1 month and a few weeks.

7.1 Future Work

Since this was the first step towards a semantic evaluation of web pages content, there are many features and fixes that should be considered and implemented in the future.

1. One of the things that should be reduced are the dependencies of third-party services. Using them turns the system vulnerable to their servers' status, because the system can not operate without theirs services, in the current state. Ideally we would have a local service that could perform semantic similarity analysis between expressions, image interpretations, generation of descriptions and text summarizers. This task is also important, because evaluation and repair can take around 30 seconds each per image. The delay happens due to the amount of calls made to these third-party services, in particular, Swoogle.

2. Currently the system only does evaluations with the English language which restrains the usage of this tool to English web pages. Despite being the most spoken language, it's not ideal to restrict the access for the ones that don't speak it. To

this end, a translation system should be merged with Screw in order to enlarge the amount of people that can benefit from it.

3. Another issue to take into account is the evaluation with text content. The system presently is capable of doing these kind of assessments, the algorithm is the same, but they need to improve since images were the focus of this work.

4. Despite of the integration in Qualweb and plugin, there wasn't any opportunity to test with users if the accessibility increases with the combination of Qualweb + Plugin + Screw. It's important to further develop more the plugin and Qualweb for repair and make tests with visually impaired people.

5. Another issue is about the amount of requests that can be made with the plugin installed in the browser. Neither Qualweb nor Screw's server are prepared to handle multiple evaluations at once. When users change their browsing tabs, the plugin makes requests to Qualweb, however the evaluations can be lost between those changes. It's necessary to create a more resilient server that can handle concurrency.

6. As was said, the plugin only works on Google Chrome, so it would be ideal having another one that can work with Internet Explorer, since it's the most used browser by visually impaired people.

7. It's also important to improve the quality of Screw's evaluations. Even if the correlation is quite positive there is space to work on it and get better results. Also, the last study showed that the current state of the algorithm it's not so suitable to every image and for that reason more analysis to a bigger set of images is needed.

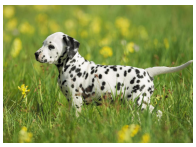
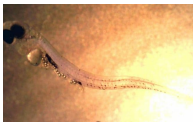
8. Introducing more domains should give more accurate results, however it needs to take into account that there are some domains stronger than others that can improve or not the results.

9. Finally, from the results of the last study, the repair module should be improved to offer better repair sentences by getting better algorithms and services that can generate those sentences

Appendix A

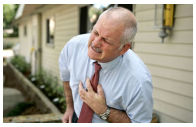
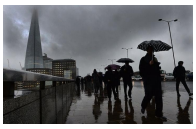
Images from survey 1

Table A.1: Correlation results

	Image	Alternative Text
1		Liberian and United States national flags with similarities being flown on a roadside in downtown Monrovia, Liberia
2		Dalmatians suffer from Hyperuricemia due to in-breeding
3		A Southern rail train
4		Watson (Martin Freeman) and Holmes (Benedict Cumberbatch) in BBC One's Sherlock
5		A larval perch that has ingested microplastic particles
6		iCloud
7		A dogfish shark
8		Theresa May delivers a major speech on Brexit today
9		download

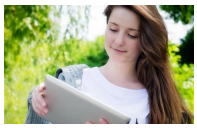
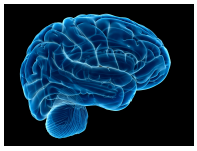
Continued on next page

Table A.1 – *Continued from previous page*

	Image	Alternative Text
10		A new app retrains the brain so that healthy food feels like a reward
11		A visitor passes over a food pellet at the Giraffe Centre in Nairobi, Kenya
12		A solar eclipse will take place on February 26 and will be visible in certain parts of the world
13		Baby crying
14		Marriage
15		The design has clear echoes of the popular red phone boxes which have become a popular part of British cityscapes
16		A man suffers a heart attack
17		Bad weather, celebrity deaths, Donald Trump and Brexit could make Blue Monday the bluest ever
18		Shaun Custis argues Usain Bolt is the greatest of all time
19		Serena Williams proved too much for Jo Konta in Melbourne
20		Jose Mourinho
21		Pair of hand-crafted gypsy wagons have been transformed into perfect holiday hideaway

Continued on next page

Table A.1 – *Continued from previous page*

	Image	Alternative Text
22		Cristiano Ronaldo insists people shouldn't compare him with Lionel Messi
23		Part of the US border with Mexico is already covered by a fence
24		In 2017, Chinese New Year falls on January 26
25		A teenager working on a tablet in the park
26		Experts think the treatment works by boosting an individual's brain power, helping them better control compulsive behaviour
27		If you want to see sunrise at Haleakala National Park in Maui, you need to book a slot
28		vegetable burger
29		A new theory about the Moon's origins has been suggested
30		Tampa skyline

Appendix B

All correlation results

Table B.1: Correlation results

thresholdsrc	zeroratio	swoogletype	form	corr
0.00	0.00	rel	max	-0.05
0.00	0.00	rel	weig	-0.05
0.00	0.00	con	max	-0.15
0.00	0.00	con	weig	-0.15
0.00	0.00	avg	max	-0.15
0.00	0.00	avg	weig	-0.15
0.00	0.10	rel	max	0.19
0.00	0.10	rel	weig	0.18
0.00	0.10	con	max	-0.02
0.00	0.10	con	weig	0.01
0.00	0.10	avg	max	0.17
0.00	0.10	avg	weig	0.20
0.00	0.20	rel	max	0.28
0.00	0.20	rel	weig	0.23
0.00	0.20	con	max	0.23
0.00	0.20	con	weig	0.28
0.00	0.20	avg	max	0.31
0.00	0.20	avg	weig	0.25
0.00	0.30	rel	max	0.31
0.00	0.30	rel	weig	0.23
0.00	0.30	con	max	0.37
0.00	0.30	con	weig	0.34
0.00	0.30	avg	max	0.33
0.00	0.30	avg	weig	0.26
0.00	0.40	rel	max	0.31
0.00	0.40	rel	weig	0.22
0.00	0.40	con	max	0.36
0.00	0.40	con	weig	0.33
0.00	0.40	avg	max	0.32
0.00	0.40	avg	weig	0.25
0.00	0.50	rel	max	0.31
0.00	0.50	rel	weig	0.22
0.00	0.50	con	max	0.36
0.00	0.50	con	weig	0.33
0.00	0.50	avg	max	0.32
0.00	0.50	avg	weig	0.25
0.00	0.60	rel	max	0.46
0.00	0.60	rel	weig	0.37
0.00	**0.60**	**con**	**max**	**0.51**
0.00	0.60	con	weig	0.48
0.00	0.60	avg	max	0.47
0.00	0.60	avg	weig	0.41
0.00	0.70	rel	max	0.46
0.00	0.70	rel	weig	0.37

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.00	**0.70**	**con**	**max**	**0.51**
0.00	0.70	con	weig	0.48
0.00	0.70	avg	max	0.47
0.00	0.70	avg	weig	0.41
0.00	0.80	rel	max	0.46
0.00	0.80	rel	weig	0.37
0.00	**0.80**	**con**	**max**	**0.51**
0.00	0.80	con	weig	0.48
0.00	0.80	avg	max	0.47
0.00	0.80	avg	weig	0.41
0.00	0.90	rel	max	0.46
0.00	0.90	rel	weig	0.37
0.00	**0.90**	**con**	**max**	**0.51**
0.00	0.90	con	weig	0.48
0.00	0.90	avg	max	0.47
0.00	0.90	avg	weig	0.41
0.00	1.00	rel	max	0.46
0.00	1.00	rel	weig	0.37
0.00	**1.00**	**con**	**max**	**0.51**
0.00	1.00	con	weig	0.48
0.00	1.00	avg	max	0.47
0.00	1.00	avg	weig	0.41
0.10	0.00	rel	max	-0.05
0.10	0.00	rel	weig	-0.05
0.10	0.00	con	max	-0.14
0.10	0.00	con	weig	-0.14
0.10	0.00	avg	max	-0.14
0.10	0.00	avg	weig	-0.15
0.10	0.10	rel	max	0.19
0.10	0.10	rel	weig	0.17
0.10	0.10	con	max	0.01
0.10	0.10	con	weig	0.01
0.10	0.10	avg	max	0.18
0.10	0.10	avg	weig	0.16
0.10	0.20	rel	max	0.32
0.10	0.20	rel	weig	0.20
0.10	0.20	con	max	0.23
0.10	0.20	con	weig	0.24
0.10	0.20	avg	max	0.32
0.10	0.20	avg	weig	0.24
0.10	0.30	rel	max	0.35
0.10	0.30	rel	weig	0.20
0.10	0.30	con	max	0.36
0.10	0.30	con	weig	0.30
0.10	0.30	avg	max	0.35
0.10	0.30	avg	weig	0.26
0.10	0.40	rel	max	0.34
0.10	0.40	rel	weig	0.19
0.10	0.40	con	max	0.35
0.10	0.40	con	weig	0.29
0.10	0.40	avg	max	0.34
0.10	0.40	avg	weig	0.26
0.10	0.50	rel	max	0.34
0.10	0.50	rel	weig	0.19
0.10	0.50	con	max	0.35
0.10	0.50	con	weig	0.29
0.10	0.50	avg	max	0.34
0.10	0.50	avg	weig	0.26
0.10	0.60	rel	max	0.50
0.10	0.60	rel	weig	0.34
0.10	0.60	con	max	0.50
0.10	0.60	con	weig	0.44
0.10	0.60	avg	max	0.49

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.10	0.60	avg	weig	0.41
0.10	0.70	rel	max	0.50
0.10	0.70	rel	weig	0.34
0.10	0.70	con	max	0.50
0.10	0.70	con	weig	0.44
0.10	0.70	avg	max	0.49
0.10	0.70	avg	weig	0.41
0.10	0.80	rel	max	0.50
0.10	0.80	rel	weig	0.34
0.10	0.80	con	max	0.50
0.10	0.80	con	weig	0.44
0.10	0.80	avg	max	0.49
0.10	0.80	avg	weig	0.41
0.10	0.90	rel	max	0.50
0.10	0.90	rel	weig	0.34
0.10	0.90	con	max	0.50
0.10	0.90	con	weig	0.44
0.10	0.90	avg	max	0.49
0.10	0.90	avg	weig	0.41
0.10	1.00	rel	max	0.50
0.10	1.00	rel	weig	0.34
0.10	1.00	con	max	0.50
0.10	1.00	con	weig	0.44
0.10	1.00	avg	max	0.49
0.10	1.00	avg	weig	0.41
0.20	0.00	rel	max	-0.05
0.20	0.00	rel	weig	-0.06
0.20	0.00	con	max	-0.14
0.20	0.00	con	weig	-0.15
0.20	0.00	avg	max	-0.14
0.20	0.00	avg	weig	-0.15
0.20	0.10	rel	max	0.18
0.20	0.10	rel	weig	0.14
0.20	0.10	con	max	0.01
0.20	0.10	con	weig	0.00
0.20	0.10	avg	max	0.17
0.20	0.10	avg	weig	0.16
0.20	0.20	rel	max	0.28
0.20	0.20	rel	weig	0.14
0.20	0.20	con	max	0.19
0.20	0.20	con	weig	0.20
0.20	0.20	avg	max	0.31
0.20	0.20	avg	weig	0.25
0.20	0.30	rel	max	0.32
0.20	0.30	rel	weig	0.16
0.20	0.30	con	max	0.32
0.20	0.30	con	weig	0.25
0.20	0.30	avg	max	0.35
0.20	0.30	avg	weig	0.29
0.20	0.40	rel	max	0.31
0.20	0.40	rel	weig	0.15
0.20	0.40	con	max	0.32
0.20	0.40	con	weig	0.24
0.20	0.40	avg	max	0.35
0.20	0.40	avg	weig	0.28
0.20	0.50	rel	max	0.31
0.20	0.50	rel	weig	0.15
0.20	0.50	con	max	0.32
0.20	0.50	con	weig	0.24
0.20	0.50	avg	max	0.35
0.20	0.50	avg	weig	0.28
0.20	0.60	rel	max	0.47
0.20	0.60	rel	weig	0.31

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.20	0.60	con	max	0.46
0.20	0.60	con	weig	0.39
0.20	0.60	avg	max	0.50
0.20	0.60	avg	weig	0.44
0.20	0.70	rel	max	0.47
0.20	0.70	rel	weig	0.31
0.20	0.70	con	max	0.46
0.20	0.70	con	weig	0.39
0.20	0.70	avg	max	0.50
0.20	0.70	avg	weig	0.44
0.20	0.80	rel	max	0.47
0.20	0.80	rel	weig	0.31
0.20	0.80	con	max	0.46
0.20	0.80	con	weig	0.39
0.20	0.80	avg	max	0.50
0.20	0.80	avg	weig	0.44
0.20	0.90	rel	max	0.47
0.20	0.90	rel	weig	0.31
0.20	0.90	con	max	0.46
0.20	0.90	con	weig	0.39
0.20	0.90	avg	max	0.50
0.20	0.90	avg	weig	0.44
0.20	1.00	rel	max	0.47
0.20	1.00	rel	weig	0.31
0.20	1.00	con	max	0.46
0.20	1.00	con	weig	0.39
0.20	1.00	avg	max	0.50
0.20	1.00	avg	weig	0.44
0.30	0.00	rel	max	-0.05
0.30	0.00	rel	weig	-0.05
0.30	0.00	con	max	-0.15
0.30	0.00	con	weig	-0.16
0.30	0.00	avg	max	-0.15
0.30	0.00	avg	weig	-0.16
0.30	0.10	rel	max	0.12
0.30	0.10	rel	weig	0.13
0.30	0.10	con	max	-0.01
0.30	0.10	con	weig	0.01
0.30	0.10	avg	max	0.18
0.30	0.10	avg	weig	0.13
0.30	0.20	rel	max	0.12
0.30	0.20	rel	weig	0.15
0.30	0.20	con	max	0.18
0.30	0.20	con	weig	0.21
0.30	0.20	avg	max	0.26
0.30	0.20	avg	weig	0.19
0.30	0.30	rel	max	0.15
0.30	0.30	rel	weig	0.16
0.30	0.30	con	max	0.30
0.30	0.30	con	weig	0.25
0.30	0.30	avg	max	0.30
0.30	0.30	avg	weig	0.22
0.30	0.40	rel	max	0.14
0.30	0.40	rel	weig	0.16
0.30	0.40	con	max	0.29
0.30	0.40	con	weig	0.25
0.30	0.40	avg	max	0.30
0.30	0.40	avg	weig	0.21
0.30	0.50	rel	max	0.14
0.30	0.50	rel	weig	0.16
0.30	0.50	con	max	0.29
0.30	0.50	con	weig	0.25
0.30	0.50	avg	max	0.30

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.30	0.50	avg	weig	0.21
0.30	0.60	rel	max	0.29
0.30	0.60	rel	weig	0.31
0.30	0.60	con	max	0.43
0.30	0.60	con	weig	0.39
0.30	0.60	avg	max	0.45
0.30	0.60	avg	weig	0.36
0.30	0.70	rel	max	0.29
0.30	0.70	rel	weig	0.31
0.30	0.70	con	max	0.43
0.30	0.70	con	weig	0.39
0.30	0.70	avg	max	0.45
0.30	0.70	avg	weig	0.36
0.30	0.80	rel	max	0.29
0.30	0.80	rel	weig	0.31
0.30	0.80	con	max	0.43
0.30	0.80	con	weig	0.39
0.30	0.80	avg	max	0.45
0.30	0.80	avg	weig	0.36
0.30	0.90	rel	max	0.29
0.30	0.90	rel	weig	0.31
0.30	0.90	con	max	0.43
0.30	0.90	con	weig	0.39
0.30	0.90	avg	max	0.45
0.30	0.90	avg	weig	0.36
0.30	1.00	rel	max	0.29
0.30	1.00	rel	weig	0.31
0.30	1.00	con	max	0.43
0.30	1.00	con	weig	0.39
0.30	1.00	avg	max	0.45
0.30	1.00	avg	weig	0.36
0.40	0.00	rel	max	-0.05
0.40	0.00	rel	weig	-0.05
0.40	0.00	con	max	-0.15
0.40	0.00	con	weig	-0.16
0.40	0.00	avg	max	-0.15
0.40	0.00	avg	weig	-0.16
0.40	0.10	rel	max	0.09
0.40	0.10	rel	weig	0.14
0.40	0.10	con	max	-0.01
0.40	0.10	con	weig	-0.01
0.40	0.10	avg	max	0.13
0.40	0.10	avg	weig	0.19
0.40	0.20	rel	max	0.14
0.40	0.20	rel	weig	0.20
0.40	0.20	con	max	0.14
0.40	0.20	con	weig	0.25
0.40	0.20	avg	max	0.18
0.40	0.20	avg	weig	0.21
0.40	0.30	rel	max	0.15
0.40	0.30	rel	weig	0.22
0.40	0.30	con	max	0.21
0.40	0.30	con	weig	0.30
0.40	0.30	avg	max	0.20
0.40	0.30	avg	weig	0.24
0.40	0.40	rel	max	0.15
0.40	0.40	rel	weig	0.21
0.40	0.40	con	max	0.21
0.40	0.40	con	weig	0.29
0.40	0.40	avg	max	0.20
0.40	0.40	avg	weig	0.23
0.40	0.50	rel	max	0.15
0.40	0.50	rel	weig	0.21

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.40	0.50	con	max	0.21
0.40	0.50	con	weig	0.29
0.40	0.50	avg	max	0.20
0.40	0.50	avg	weig	0.23
0.40	0.60	rel	max	0.29
0.40	0.60	rel	weig	0.36
0.40	0.60	con	max	0.34
0.40	0.60	con	weig	0.43
0.40	0.60	avg	max	0.33
0.40	0.60	avg	weig	0.38
0.40	0.70	rel	max	0.29
0.40	0.70	rel	weig	0.36
0.40	0.70	con	max	0.34
0.40	0.70	con	weig	0.43
0.40	0.70	avg	max	0.33
0.40	0.70	avg	weig	0.38
0.40	0.80	rel	max	0.29
0.40	0.80	rel	weig	0.36
0.40	0.80	con	max	0.34
0.40	0.80	con	weig	0.43
0.40	0.80	avg	max	0.33
0.40	0.80	avg	weig	0.38
0.40	0.90	rel	max	0.29
0.40	0.90	rel	weig	0.36
0.40	0.90	con	max	0.34
0.40	0.90	con	weig	0.43
0.40	0.90	avg	max	0.33
0.40	0.90	avg	weig	0.38
0.40	1.00	rel	max	0.29
0.40	1.00	rel	weig	0.36
0.40	1.00	con	max	0.34
0.40	1.00	con	weig	0.43
0.40	1.00	avg	max	0.33
0.40	1.00	avg	weig	0.38
0.50	0.00	rel	max	-0.06
0.50	0.00	rel	weig	-0.05
0.50	0.00	con	max	-0.15
0.50	0.00	con	weig	-0.15
0.50	0.00	avg	max	-0.15
0.50	0.00	avg	weig	-0.16
0.50	0.10	rel	max	0.18
0.50	0.10	rel	weig	0.15
0.50	0.10	con	max	-0.00
0.50	0.10	con	weig	0.01
0.50	0.10	avg	max	0.18
0.50	0.10	avg	weig	0.20
0.50	0.20	rel	max	0.27
0.50	0.20	rel	weig	0.20
0.50	0.20	con	max	0.10
0.50	0.20	con	weig	0.24
0.50	0.20	avg	max	0.15
0.50	0.20	avg	weig	0.24
0.50	0.30	rel	max	0.31
0.50	0.30	rel	weig	0.24
0.50	0.30	con	max	0.14
0.50	0.30	con	weig	0.28
0.50	0.30	avg	max	0.18
0.50	0.30	avg	weig	0.28
0.50	0.40	rel	max	0.30
0.50	0.40	rel	weig	0.23
0.50	0.40	con	max	0.14
0.50	0.40	con	weig	0.27
0.50	0.40	avg	max	0.18

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.50	0.40	avg	weig	0.27
0.50	0.50	rel	max	0.30
0.50	0.50	rel	weig	0.23
0.50	0.50	con	max	0.14
0.50	0.50	con	weig	0.27
0.50	0.50	avg	max	0.18
0.50	0.50	avg	weig	0.27
0.50	0.60	rel	max	0.44
0.50	0.60	rel	weig	0.38
0.50	0.60	con	max	0.26
0.50	0.60	con	weig	0.41
0.50	0.60	avg	max	0.31
0.50	0.60	avg	weig	0.41
0.50	0.70	rel	max	0.44
0.50	0.70	rel	weig	0.38
0.50	0.70	con	max	0.26
0.50	0.70	con	weig	0.41
0.50	0.70	avg	max	0.31
0.50	0.70	avg	weig	0.41
0.50	0.80	rel	max	0.44
0.50	0.80	rel	weig	0.38
0.50	0.80	con	max	0.26
0.50	0.80	con	weig	0.41
0.50	0.80	avg	max	0.31
0.50	0.80	avg	weig	0.41
0.50	0.90	rel	max	0.44
0.50	0.90	rel	weig	0.38
0.50	0.90	con	max	0.26
0.50	0.90	con	weig	0.41
0.50	0.90	avg	max	0.31
0.50	0.90	avg	weig	0.41
0.50	1.00	rel	max	0.44
0.50	1.00	rel	weig	0.38
0.50	1.00	con	max	0.26
0.50	1.00	con	weig	0.41
0.50	1.00	avg	max	0.31
0.50	1.00	avg	weig	0.41
0.60	0.00	rel	max	-0.05
0.60	0.00	rel	weig	-0.06
0.60	0.00	con	max	-0.16
0.60	0.00	con	weig	-0.15
0.60	0.00	avg	max	-0.15
0.60	0.00	avg	weig	-0.15
0.60	0.10	rel	max	0.16
0.60	0.10	rel	weig	0.15
0.60	0.10	con	max	-0.03
0.60	0.10	con	weig	-0.01
0.60	0.10	avg	max	0.18
0.60	0.10	avg	weig	0.21
0.60	0.20	rel	max	0.27
0.60	0.20	rel	weig	0.19
0.60	0.20	con	max	0.03
0.60	0.20	con	weig	0.19
0.60	0.20	avg	max	0.11
0.60	0.20	avg	weig	0.19
0.60	0.30	rel	max	0.31
0.60	0.30	rel	weig	0.23
0.60	0.30	con	max	0.09
0.60	0.30	con	weig	0.23
0.60	0.30	avg	max	0.15
0.60	0.30	avg	weig	0.23
0.60	0.40	rel	max	0.30
0.60	0.40	rel	weig	0.22

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.60	0.40	con	max	0.09
0.60	0.40	con	weig	0.22
0.60	0.40	avg	max	0.14
0.60	0.40	avg	weig	0.22
0.60	0.50	rel	max	0.30
0.60	0.50	rel	weig	0.22
0.60	0.50	con	max	0.09
0.60	0.50	con	weig	0.22
0.60	0.50	avg	max	0.14
0.60	0.50	avg	weig	0.22
0.60	0.60	rel	max	0.43
0.60	0.60	rel	weig	0.36
0.60	0.60	con	max	0.22
0.60	0.60	con	weig	0.35
0.60	0.60	avg	max	0.27
0.60	0.60	avg	weig	0.36
0.60	0.70	rel	max	0.43
0.60	0.70	rel	weig	0.36
0.60	0.70	con	max	0.22
0.60	0.70	con	weig	0.35
0.60	0.70	avg	max	0.27
0.60	0.70	avg	weig	0.36
0.60	0.80	rel	max	0.43
0.60	0.80	rel	weig	0.36
0.60	0.80	con	max	0.22
0.60	0.80	con	weig	0.35
0.60	0.80	avg	max	0.27
0.60	0.80	avg	weig	0.36
0.60	0.90	rel	max	0.43
0.60	0.90	rel	weig	0.36
0.60	0.90	con	max	0.22
0.60	0.90	con	weig	0.35
0.60	0.90	avg	max	0.27
0.60	0.90	avg	weig	0.36
0.60	1.00	rel	max	0.43
0.60	1.00	rel	weig	0.36
0.60	1.00	con	max	0.22
0.60	1.00	con	weig	0.35
0.60	1.00	avg	max	0.27
0.60	1.00	avg	weig	0.36
0.70	0.00	rel	max	-0.05
0.70	0.00	rel	weig	-0.06
0.70	0.00	con	max	-0.16
0.70	0.00	con	weig	-0.16
0.70	0.00	avg	max	-0.15
0.70	0.00	avg	weig	-0.15
0.70	0.10	rel	max	0.17
0.70	0.10	rel	weig	0.16
0.70	0.10	con	max	-0.03
0.70	0.10	con	weig	-0.02
0.70	0.10	avg	max	0.13
0.70	0.10	avg	weig	0.20
0.70	0.20	rel	max	0.26
0.70	0.20	rel	weig	0.28
0.70	0.20	con	max	0.07
0.70	0.20	con	weig	0.22
0.70	0.20	avg	max	0.05
0.70	0.20	avg	weig	0.20
0.70	0.30	rel	max	0.29
0.70	0.30	rel	weig	0.32
0.70	0.30	con	max	0.14
0.70	0.30	con	weig	0.27
0.70	0.30	avg	max	0.08

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.70	0.30	avg	weig	0.25
0.70	0.40	rel	max	0.29
0.70	0.40	rel	weig	0.32
0.70	0.40	con	max	0.14
0.70	0.40	con	weig	0.26
0.70	0.40	avg	max	0.07
0.70	0.40	avg	weig	0.24
0.70	0.50	rel	max	0.29
0.70	0.50	rel	weig	0.32
0.70	0.50	con	max	0.14
0.70	0.50	con	weig	0.26
0.70	0.50	avg	max	0.07
0.70	0.50	avg	weig	0.24
0.70	0.60	rel	max	0.42
0.70	0.60	rel	weig	0.46
0.70	0.60	con	max	0.26
0.70	0.60	con	weig	0.39
0.70	0.60	avg	max	0.20
0.70	0.60	avg	weig	0.37
0.70	0.70	rel	max	0.42
0.70	0.70	rel	weig	0.46
0.70	0.70	con	max	0.26
0.70	0.70	con	weig	0.39
0.70	0.70	avg	max	0.20
0.70	0.70	avg	weig	0.37
0.70	0.80	rel	max	0.42
0.70	0.80	rel	weig	0.46
0.70	0.80	con	max	0.26
0.70	0.80	con	weig	0.39
0.70	0.80	avg	max	0.20
0.70	0.80	avg	weig	0.37
0.70	0.90	rel	max	0.42
0.70	0.90	rel	weig	0.46
0.70	0.90	con	max	0.26
0.70	0.90	con	weig	0.39
0.70	0.90	avg	max	0.20
0.70	0.90	avg	weig	0.37
0.70	1.00	rel	max	0.42
0.70	1.00	rel	weig	0.46
0.70	1.00	con	max	0.26
0.70	1.00	con	weig	0.39
0.70	1.00	avg	max	0.20
0.70	1.00	avg	weig	0.37
0.80	0.00	rel	max	-0.06
0.80	0.00	rel	weig	-0.06
0.80	0.00	con	max	-0.16
0.80	0.00	con	weig	-0.15
0.80	0.00	avg	max	-0.15
0.80	0.00	avg	weig	-0.15
0.80	0.10	rel	max	0.10
0.80	0.10	rel	weig	0.18
0.80	0.10	con	max	-0.03
0.80	0.10	con	weig	-0.01
0.80	0.10	avg	max	0.15
0.80	0.10	avg	weig	0.23
0.80	0.20	rel	max	0.13
0.80	0.20	rel	weig	0.30
0.80	0.20	con	max	0.08
0.80	0.20	con	weig	0.17
0.80	0.20	avg	max	0.12
0.80	0.20	avg	weig	0.23
0.80	0.30	rel	max	0.17
0.80	0.30	rel	weig	0.34

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.80	0.30	con	max	0.16
0.80	0.30	con	weig	0.23
0.80	0.30	avg	max	0.16
0.80	0.30	avg	weig	0.28
0.80	0.40	rel	max	0.16
0.80	0.40	rel	weig	0.34
0.80	0.40	con	max	0.15
0.80	0.40	con	weig	0.23
0.80	0.40	avg	max	0.15
0.80	0.40	avg	weig	0.27
0.80	0.50	rel	max	0.16
0.80	0.50	rel	weig	0.34
0.80	0.50	con	max	0.15
0.80	0.50	con	weig	0.23
0.80	0.50	avg	max	0.15
0.80	0.50	avg	weig	0.27
0.80	0.60	rel	max	0.28
0.80	0.60	rel	weig	0.46
0.80	0.60	con	max	0.26
0.80	0.60	con	weig	0.35
0.80	0.60	avg	max	0.27
0.80	0.60	avg	weig	0.40
0.80	0.70	rel	max	0.28
0.80	0.70	rel	weig	0.46
0.80	0.70	con	max	0.26
0.80	0.70	con	weig	0.35
0.80	0.70	avg	max	0.27
0.80	0.70	avg	weig	0.40
0.80	0.80	rel	max	0.28
0.80	0.80	rel	weig	0.46
0.80	0.80	con	max	0.26
0.80	0.80	con	weig	0.35
0.80	0.80	avg	max	0.27
0.80	0.80	avg	weig	0.40
0.80	0.90	rel	max	0.28
0.80	0.90	rel	weig	0.46
0.80	0.90	con	max	0.26
0.80	0.90	con	weig	0.35
0.80	0.90	avg	max	0.27
0.80	0.90	avg	weig	0.40
0.80	1.00	rel	max	0.28
0.80	1.00	rel	weig	0.46
0.80	1.00	con	max	0.26
0.80	1.00	con	weig	0.35
0.80	1.00	avg	max	0.27
0.80	1.00	avg	weig	0.40
0.90	0.00	rel	max	-0.06
0.90	0.00	rel	weig	-0.06
0.90	0.00	con	max	-0.16
0.90	0.00	con	weig	-0.16
0.90	0.00	avg	max	-0.15
0.90	0.00	avg	weig	-0.15
0.90	0.10	rel	max	0.05
0.90	0.10	rel	weig	0.15
0.90	0.10	con	max	-0.05
0.90	0.10	con	weig	-0.05
0.90	0.10	avg	max	0.16
0.90	0.10	avg	weig	0.20
0.90	0.20	rel	max	0.05
0.90	0.20	rel	weig	0.21
0.90	0.20	con	max	-0.04
0.90	0.20	con	weig	0.03
0.90	0.20	avg	max	0.17

Continued on next page

Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
0.90	0.20	avg	weig	0.20
0.90	0.30	rel	max	0.07
0.90	0.30	rel	weig	0.24
0.90	0.30	con	max	0.01
0.90	0.30	con	weig	0.07
0.90	0.30	avg	max	0.21
0.90	0.30	avg	weig	0.23
0.90	0.40	rel	max	0.07
0.90	0.40	rel	weig	0.24
0.90	0.40	con	max	0.01
0.90	0.40	con	weig	0.07
0.90	0.40	avg	max	0.20
0.90	0.40	avg	weig	0.23
0.90	0.50	rel	max	0.07
0.90	0.50	rel	weig	0.24
0.90	0.50	con	max	0.01
0.90	0.50	con	weig	0.07
0.90	0.50	avg	max	0.20
0.90	0.50	avg	weig	0.23
0.90	0.60	rel	max	0.16
0.90	0.60	rel	weig	0.35
0.90	0.60	con	max	0.09
0.90	0.60	con	weig	0.16
0.90	0.60	avg	max	0.32
0.90	0.60	avg	weig	0.33
0.90	0.70	rel	max	0.16
0.90	0.70	rel	weig	0.35
0.90	0.70	con	max	0.09
0.90	0.70	con	weig	0.16
0.90	0.70	avg	max	0.32
0.90	0.70	avg	weig	0.33
0.90	0.80	rel	max	0.16
0.90	0.80	rel	weig	0.35
0.90	0.80	con	max	0.09
0.90	0.80	con	weig	0.16
0.90	0.80	avg	max	0.32
0.90	0.80	avg	weig	0.33
0.90	0.90	rel	max	0.16
0.90	0.90	rel	weig	0.35
0.90	0.90	con	max	0.09
0.90	0.90	con	weig	0.16
0.90	0.90	avg	max	0.32
0.90	0.90	avg	weig	0.33
0.90	1.00	rel	max	0.16
0.90	1.00	rel	weig	0.35
0.90	1.00	con	max	0.09
0.90	1.00	con	weig	0.16
0.90	1.00	avg	max	0.32
0.90	1.00	avg	weig	0.33
1.00	0.00	rel	max	-0.06
1.00	0.00	rel	weig	-0.06
1.00	0.00	con	max	-0.16
1.00	0.00	con	weig	-0.17
1.00	0.00	avg	max	-0.16
1.00	0.00	avg	weig	-0.16
1.00	0.10	rel	max	0.02
1.00	0.10	rel	weig	0.09
1.00	0.10	con	max	-0.04
1.00	0.10	con	weig	-0.07
1.00	0.10	avg	max	0.11
1.00	0.10	avg	weig	0.18
1.00	0.20	rel	max	0.01
1.00	0.20	rel	weig	0.15

Continued on next page

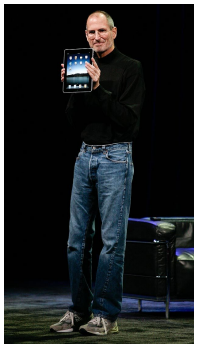
Table B.1 – *Continued from previous page*

thresholdsrc	zeroratio	swoogletype	form	corr
1.00	0.20	con	max	0.01
1.00	0.20	con	weig	-0.03
1.00	0.20	avg	max	0.11
1.00	0.20	avg	weig	0.20
1.00	0.30	rel	max	0.04
1.00	0.30	rel	weig	0.19
1.00	0.30	con	max	0.09
1.00	0.30	con	weig	0.01
1.00	0.30	avg	max	0.14
1.00	0.30	avg	weig	0.23
1.00	0.40	rel	max	0.03
1.00	0.40	rel	weig	0.18
1.00	0.40	con	max	0.08
1.00	0.40	con	weig	-0.00
1.00	0.40	avg	max	0.13
1.00	0.40	avg	weig	0.22
1.00	0.50	rel	max	0.03
1.00	0.50	rel	weig	0.18
1.00	0.50	con	max	0.08
1.00	0.50	con	weig	-0.00
1.00	0.50	avg	max	0.13
1.00	0.50	avg	weig	0.22
1.00	0.60	rel	max	0.12
1.00	0.60	rel	weig	0.28
1.00	0.60	con	max	0.16
1.00	0.60	con	weig	0.08
1.00	0.60	avg	max	0.22
1.00	0.60	avg	weig	0.31
1.00	0.70	rel	max	0.12
1.00	0.70	rel	weig	0.28
1.00	0.70	con	max	0.16
1.00	0.70	con	weig	0.08
1.00	0.70	avg	max	0.22
1.00	0.70	avg	weig	0.31
1.00	0.80	rel	max	0.12
1.00	0.80	rel	weig	0.28
1.00	0.80	con	max	0.16
1.00	0.80	con	weig	0.08
1.00	0.80	avg	max	0.22
1.00	0.80	avg	weig	0.31
1.00	0.90	rel	max	0.12
1.00	0.90	rel	weig	0.28
1.00	0.90	con	max	0.16
1.00	0.90	con	weig	0.08
1.00	0.90	avg	max	0.22
1.00	0.90	avg	weig	0.31
1.00	1.00	rel	max	0.12
1.00	1.00	rel	weig	0.28
1.00	1.00	con	max	0.16
1.00	1.00	con	weig	0.08
1.00	1.00	avg	max	0.22
1.00	1.00	avg	weig	0.31

Appendix C

Images from survey 2

Table C.1: Second survey descriptions

	image	original description	captionbot description	concepts description
1		This week our skies have some shooting stars – but mainly during daylight hours	A close up of a rock next to a body of water.	This image is mainly about these concepts: seashore, landscape and sea
2		An illustration of the skull of the most ancient human ever discovered	A close up of an animal.	This image is mainly about these concepts: skull, frame and bone
3		Could this poor porker's lugs help to cure a selfish human's hangover?	A close up of an animal.	This image is mainly about these concepts: mammal, farm and livestock
4		The trainers are selling on eBay at a starting bid of £15,000 or £12,000	A close up of a footwear.	This image is mainly about these concepts: sneakers, isolated and fashion
5		Apple Inc. CEO Steve Jobs	Steve Jobs standing in a dark room,and he seems to have a grinning face with smiling eyes.	This image is mainly about these concepts: man, people and portrait

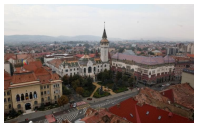
Continued on next page

Table C.1 – *Continued from previous page*

	image	original description	captionbot description	concepts description
6		Adam watched helplessly as his car burnt to a cinder	A car parked on the side of a road.	This image is mainly about these concepts: car, accident and vehicle
7		Joust the job… Warwick Castle had plenty of action on show	A group of people riding on the back of a horse.	This image is mainly about these concepts: festival, people and competition
8		Hard Rock Hotel Tenerife	A boat parked on the side of a building.	This image is mainly about these concepts: water, travel and no person
9		Uber confirmed Eric Alexander's departure	A close up of person.	This image is mainly about these concepts: business, vehicle and people
10		Donald Trump fired James Comey, right, as head of the FBI last month	Donald Trump, James Comey are posing for a picture,and they seem to have a neutral face.	This image is mainly about these concepts: portrait, politician and one
11		cat on a leash	A dog sitting on top of a grass covered field.	This image is mainly about these concepts: grass, field and hayfield
12		little girl and her dog sitting together on a couch	A brown and white dog lying on a couch.	This image is mainly about these concepts: one, dog and people
13		Motus MSTR studio side view	A motorcycle parked on the side of a road.	This image is mainly about these concepts: bike, wheel and vehicle

Continued on next page

Table C.1 – *Continued from previous page*

	image	original description	captionbot description	concepts description
14		Sunday Telegraph In Romania with Colin Freeman for story on immigration rules to the UK for Romanian (and Bulgarian) nationals	A large building with a mountain in the background.	This image is mainly about these concepts: roof, architecture and city
15		Fruits and vegetables are presented during the opening day of the Fruit Logistica trade fair in Berlin	A group of fruit and vegetables on display.	This image is mainly about these concepts: market, food and fruit

Bibliography

[1] About wordnet - wordnet - about wordnet. https://wordnet.princeton.edu/wordnet/. Accessed on 2016-12-1.

[2] Accessibility testing. https://www.w3.org/wiki/Accessibility_testing/. Accessed on 2017-07-28.

[3] Accessible rich internet applications. https://www.w3.org/TR/wai-aria/. Accessed on 2016-11-24.

[4] Accessible rich internet applications (wai-aria) 1.0 - rich internet application accessibility. https://www.w3.org/TR/wai-aria/introduction#intro_ria_accessibility. Accessed on 2016-11-27.

[5] Javascript usage statistics. https://trends.builtwith.com/docinfo/Javascript. Accessed on 2016-22-12.

[6] Understanding conformance. https://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html. Accessed on 2016-11-24.

[7] Understanding the four principles of accessibility. https://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html#introduction-fourprincs-head. Accessed on 2016-11-24.

[8] Web acessibility initiative. https://www.w3.org/WAI/. Accessed on 2016-11-07.

[9] Web acessibility initiative. https://www.w3.org/WAI/WCAG20/from10/diff. Accessed on 2017-08-25.

[10] Web content accessibility guidelines. https://www.w3.org/TR/WCAG20/. Accessed on 2016-11-07.

[11] World wide web consortium. https://www.w3.org/. Accessed on 2016-11-07.

[12] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[13] Chieko Asakawa and Hironobu Takagi. Annotation-based transcoding for nonvisual web access. In *Proceedings of the fourth international ACM conference on Assistive technologies*, pages 172–179. ACM, 2000.

[14] Chieko Asakawa and Hironobu Takagi. Transcoding. In *Web Accessibility*, pages 231–260. Springer, 2008.

[15] Chris Biemann. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93, 2005.

[16] Michael C Burl, Charless Fowlkes, and Joseph Roden. Mining for image content. *Systemics, cybernetics, and informatics/information systems: analysis and synthesis*, page 9, 1999.

[17] Antonio Carzaniga, Alessandra Gorla, Nicolò Perino, and Mauro Pezzè. Automatic workarounds for web applications. In *Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering*, pages 237–246. ACM, 2010.

[18] World Wide Web Consortium et al. Techniques for accessibility evaluation and repair tools. *http://www. w3. org/TR/2000/WD-AERT-20000426*, 2000.

[19] Michael Cooper. Accessibility of emerging rich web technologies: web 2.0 and the semantic web. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, pages 93–98. ACM, 2007.

[20] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1777–1784. IEEE, 2011.

[21] Nádia Fernandes. Towards web accessibility repair. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, page 9. ACM, 2013.

[22] Nádia Fernandes, Ana Sofia Batista, Daniel Costa, Carlos Duarte, and Luís Carriço. Three web accessibility evaluation perspectives for ria. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, page 12. ACM, 2013.

[23] Nádia Fernandes and Luís Carriço. Assessing the effort of repairing the accessibility of web sites. In *International Conference on Computers for Handicapped Persons*, pages 396–403. Springer, 2012.

[24] Nádia Fernandes, Daniel Costa, Sergio Neves, Carlos Duarte, and Luís Carriço. Evaluating the accessibility of rich internet applications. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*, page 13. ACM, 2012.

[25] Lushan Han, Abhay L Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc_ebiquity-core: Semantic textual similarity systems. In ** SEM@ NAACL-HLT*, pages 44–52, 2013.

[26] Chandra Harrison and Helen Petrie. Severity of usability and accessibility problems in ecommerce and egovernment websites. In *People and Computers XX—Engage*, pages 255–262. Springer, 2007.

[27] Ian Horrocks and Sean Bechhofer. Semantic web. Human-Computer Interaction Series, chapter 19, pages 315–330. Springer, London, 1st edition, September 2008.

[28] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[29] Eduard Klein, Anton Bolfing, and Markus Riesch. Checking web accessibility with the content accessibility checker (cac). In *International Conference on Computers for Handicapped Persons*, pages 109–112. Springer, 2014.

[30] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[31] Chee Wee Leong and Rada Mihalcea. Measuring the semantic relatedness between words and images. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 185–194. Association for Computational Linguistics, 2011.

[32] Sergio Luján-Mora. A comparison of common web accessibility problems. `http://desarrolloweb.dlsi.ua.es/web-accessibility/comparison-common-web-accessibility-problems`. Accessed on 2017-09-11.

[33] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[34] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[35] Martin Monperrus. Automatic Software Repair: a Bibliography. Technical Report hal-01206501, University of Lille, 2015.

[36] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[37] Joel Larocca Neto, Alexandre D. Santos, Celso A.A. Kaestner, and Alex A. Freitas. Document clustering and text summarization, 2000.

[38] Christopher Power, Helen Petrie, André P Freire, and David Swallow. Remote evaluation of wcag 2.0 techniques by web users with visual disabilities. In *International Conference on Universal Access in Human-Computer Interaction*, pages 285–294. Springer, 2011.

[39] Loretta Guarino Reid and Andi Snow-Weaver. Wcag 2.0: a web accessibility standard for the evolving web. In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*, pages 109–115. ACM, 2008.

[40] André Rodrigues and Tiago Guerreiro. Swat: Mobile system-wide assistive technologies. In *Proceedings of the 28th International BCS Human Computer Interaction Conference on HCI 2014-Sand, Sea and Sky-Holiday HCI*, pages 341–346. BCS, 2014.

[41] Sandra Souza Rodrigues, Renata Pontin de Mattos Fortes, and André Pimenta Freire. Towards characteristics of accessibility and usability issues for older people-a brazilian case study. In *International Conference on Human Aspects of IT for the Aged Population*, pages 117–128. Springer, 2016.

[42] Dagfinn Rømen and Dag Svanæs. Validating wcag versions 1.0 and 2.0 through usability testing with disabled users. *Universal Access in the Information Society*, 11(4):375–385, 2012.

[43] Deborah J. Rumsey. How to interpret a correlation coefficient r. `http://www.dummies.com/education/math/statistics/how-to-interpret-a-correlation-coefficient-r/`. Accessed on 2017-07-16.

[44] Sergio Sayago, Laura Camacho, and Josep Blat. Evaluation of techniques defined in wcag 2.0 with older people. In *Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A)*, pages 79–82. ACM, 2009.

[45] Clare M. So, Mark Perry, and Stephen M. Watt. Towards an accessible web through semantic web standards. In *Proceedings of The 2005 International Conference on Computers for People with Special Needs, CPSN 2005, Las Vegas, Nevada, USA, June 20-23, 2005*, pages 10–16, 2005.

[46] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

[47] Internet Live Stats. Internet users. `http://www.internetlivestats.com/internet-users/`. Accessed on 2016-10-24.

[48] Weihang Wang, Yunhui Zheng, Peng Liu, Lei Xu, Xiangyu Zhang, and Patrick Eugster. Arrow: automated repair of races on client-side web pages. In *Proceedings of the 25th International Symposium on Software Testing and Analysis*, pages 201–212. ACM, 2016.

[49] WB WHO. World report on disability. *Geneva: WHO*, 2011.

[50] Anna Wierzbicka. The meaning of color terms: semantics, culture, and cognition. *Cognitive Linguistics (includes Cognitive Linguistic Bibliography)*, 1(1):99–150, 1990.

[51] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[52] Ji Zhang, Wynne Hsu, and Mong Li Lee. Image mining: Issues, frameworks and techniques. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Multimedia Data Mining (MDM/KDD'01)*. University of Alberta, 2001.